HENDRIX 6400 COMPOSITION MARK-UP MANUAL

9-03878-01
July, 1977

COLLOPHON

This manual was keyboarded, formatted, edited and revised on a Hendrix 6480 Text Publishing System and output to an on-line CRT phototypesetter.

The main text is set in UNIVERS 55 (Medium) with 75 (Extra Bold) plus variations produced by the typesetter itself. Text examples shown are in Corona style.

PROPRIETARY NOTICE

All information contained in this manual is the exclusive property of Hendrix Wordwide, Inc. and shall not be copied or reproduced in any form without the expressed written consent of:

HENDRIX
WORLDWIDE

TABLE OF CONTENTS

Section I Introduction to Mark-up

1.1 About this Manual....1-1
1.2 Mark-Up and Where it's Done....1-1
1.3 The 6400 Text Publishing System....1-2
1.4 Other Publications....1-3

Section II Command Structure

2.1 Introduction....2-1
2.2 Terminology....2-1
2.3 Single Key Commands....2-2
2.4 2-Letter Function Commands....2-4
2.5 Elements of a 6400 command....2-7
2.6 Command Separators....2-8
2.7 Error Messages....2-10

Section III Composition Commands

3.1 Introduction....3-1
3.2 How to use the command pages....3-2
3.3 Style and Size Control Group....3-3
3.4 Leading and Spacing Control Group....3-16
3.5 Hyphenation and Justification Control Group....3-21
3.6 Line Control Group....3-37
3.7 Indent Control Group....3-43
3.8 Tabbing Control Group....3-54
3.9 Compose-for-Display Group....3-63
3.10 Miscellaneous Commands Group....3-69
3.11 Cancel Commands Group....3-74
3.12 Special Typesetter Commands Group....3-75

Section IV Formats and Macros

4.1 Introduction 4-1
4.2 What is a Format 4-2
4.3 What is a Macro 4-2
4.4 What is a General Variable 4-2
4.5 What is a Merge 4-2
4.6 Format Commands 4-4
4.7 Macro Commands 4-6
4.8 Merge Commands 4-8
4.9 String Commands 4-10
4.10 General Variables 4-12

PREFACE

THIS MARK-UP MANUAL introduces you to the 6400 Hendrix Composition System, known as HCS (pronounced "hicks"). The information contained herein can only, first guide, then later reinforce your path to proficiency. Usage itself will finally prove to be the best instructor.

THIS MARK-UP MANUAL was designed to serve as a handy reference tool as well as a basic instructional book. The important facts or codes are in **bold type** or in BIG type, for easy referral. By far, the most important feature is that for EVERY command there is an illustrative example and it's on the SAME PAGE.

THIS MARK-UP MANUAL itself demonstrates the capabilities of HCS. Every command is shown in action. The power of formats and macros was used extensively. Their usage simplified keyboarding and reduced errors. Every page appears exactly as it emerged from the phototypesetter. Every page is a complete unit and includes rules, folios and various running heads. Paste-up was not necessary, except for the command examples, which were separate columns, complete unto themselves.

THIS MARK-UP MANUAL thus graphically displays the effective use of the commands, directions, instructions and advice that is described within its pages.

SECTION I
INTRODUCTION

1.1 About This Manual

This manual is limited in scope. It deals only with the language of the composition program. It will **not** familiarize you with any equipment or tell you what buttons to push. It **will** tell you how to achieve typographic results from the output typesetter.

There is a lot of information in this manual. It cannot be learned in one reading. It has been designed so that the pertinent facts are easy to find. The back cover contains a summary of all commands with a reference to the page where the description and examples of the command are found. Go to these pages if you are not exactly sure how a command works in a special situation.

1.2 Mark-up and Where it's Done

Traditionally, mark-up coding was performed directly on the "hard copy", using a local shorthand to indicate type style, type size, spacing, etc. The typesetting operator then interpreted these codes and performed various actions according to the dictates of his machine.

You will be using a special language (coding) for your mark-up of "soft copy" while it is still in the system. If you don't follow the rules, the system will call for help, by sending you an error message.

There are three major areas where mark up codes are used. They are:

1. **Before keyboarding** on the original copy.
2. **During keyboarding** on a terminal, or a typewriter (for later conversion by an OCR), or perhaps on a paper tape perforator.
3. **After keyboarding** on a terminal by adding commands or format calls to the existing text.

The choice depends on many factors, such as (1) the nature of the original, (2) your experience, (3) availability of equipment, (4) the typesetter and typefaces available, etc.

No matter which technique or equipment is used, the same language and coding presented in this manual is to be used.

1.3 The 6400 Text Publishing System

The Hendrix 6400 system is a computerized text publishing system which is capable of performing virtually all of the "copy handling" functions required for newspaper production. From a user's point of view, the 6400 has some major advantages over the traditional typing, editing, retyping, and tape-perforating of copy. With the Hendrix system, virgin copy is typed only once and then stored in the system. From then on, terminal operators can correct, expand, contract, rearrange and combine the copy in every imaginable way, without retyping.

Physically, the 6400 consists of a number of complex electronic components that perform highly specialized functions: however, the device most often used for copy preparation is the Video Display Terminal (VDT). Typically, copy is typed and entered into the system from the VDT in much the same way that copy is typed on a standard typewriter. However, once the copy is in the system, it can be recalled for display on a VDT screen, edited any number of times, passed electronically from one VDT to another (e.g., from a reporter to his editor), composed, and finally transmitted to a typesetter.

In many 6400 installations, the composition process (text hyphenation and justification) is accomplished right in the system by an option called the Hendrix Composition System, or HCS (pronounced "hicks"). This option not only composes the text, but also allows the user to display the composed copy on the VDT screen for verification or further editing before the text is transmitted to the typesetter. In other Hendrix installations without the HCS option, the hyphenation and justification process (h & j) is performed by an intervening computer or by the typesetter itself. In any case, the ultimate typesetter output — excluding advertisement copy — is hyphenated and justified columns of text.

SECT. I—INTRODUCTION TO MARK-UP

1.3 The 6400 Text Publishing System

By incorporating some, or all of the various 6400 hardware and software options, such as: classified-ad software, OCR input capability, direct on-line newswire storage, et. al.; the data handling capacity and system flexibility can be greatly enhanced.

1.4 Other Publications

If your job entails more than mark-up, or you desire more information about the Hendrix 6400 Text Publishing System, there are other publications that you can use. The table below and on the next page shows what is available.

Most of these documents are shipped with each system. However, some manuals may be in the process of being completed or revised. If so, they will not be shipped with your system and will be shipped at a later date. Any publication which does not appear on the shipping manifesto does not come with your system, but may be purchased separately by using the part number given in the last column.

6400 System Reference Documents

Publisher	System or Device	Type of Manual	Part Number
Hendrix	6400 Text Publishing System	Operators Manual	9-03828-01
Hendrix	6400 Editorial	Users Manual	9-01942-02
Hendrix	6400 Classified	Users Manual	9-03643-01
Hendrix	6400 Composition	Users Manual	9-04239-01
Hendrix	6400 Composition	Mark-up Manual	9-03878-01
Hendrix	6400 Text Publishing System	Technical/Maintenance Manual	9-02242-01
Hendrix	6400 Text Publishing System	Engineering Drawings and Schematics	9-02243-01
Hendrix	6400 Text Publishing System	Diagnostic Package	5-00078-02
Hendrix	6400 Text Publishing System	Utilities Program Package	5-00073-01
Hendrix	6400 Application Notes	Bulletin	9-02478-01
Hendrix	11/34 and 11/04 Computer	Reference Manual	9-03879-01
Hendrix	Facilities Planning	Manual	9-03009-01-A

continued on next page

1.4 Other publications

6400 System Reference Documents (continued)

Publisher	System or Device	Type of Manual	Part Number
DEC	861 Power Controller	Maintenance Manual	1-00514-01
DEC	861 Power Controller	Engineering Drawings	1-00520-01
DEC	RK05 Disk Drive	Maintenance Manual	1-00512-01
DEC	RK05 Disk Drive	Engineering Drawings	1-00516-01
DEC	RK05 Disk Drive	Illustrated Parts Breakdown	1-00522-01
DEC	RK11D/E Disk Drive Controller	Engineering Drawings	1-00517-01
DEC	KD11-A Processor	Maintenance Manual	1-00511-01
DEC	BA11-K Mounting Box	Manual	1-00521-01
DEC	PDP-11 Peripherals	Handbook	1-16405-01
DEC	PDP-11 Computer	Programming Card	1-16406-02
DEC	H720 Power Supply	Manual	1-00526-01
DEC	H720 Power Supply	Engineering Drawings	1-00519-01
DEC	H765 Power System and Mounting Box	Engineering Drawings	1-00518-01
Plessy	PM1116 Memory	Manual	1-16209-01
Plessy	PM1105 Memory	Manual	1-16210-01
Teletype	KSR-33 Vols 1 and 2	Technical Manual	1-16216-01/02
Teletype	KSR-33	Parts Manual	1-16217-01
Teletype	KSR-33	Wiring Diagram Package	1-16212-01
Facit	4070 Tape Punch	Service Manual	1-16218-01
Decitek	241, 242, 243 Basic Tape Reader	Instruction Manual	1-16211-01
Motorola	Display Monitor	Technical Repair Manual	1-00515-01

SECTION II
COMMAND STRUCTURE

2.1 Introduction

A phototypsetter recieves a data stream of coded signals which it immediately separates into one of two classes. Either the code stands for:

(1) **A character** to be flashed or generated and thus exposed onto paper or

(2) **An instruction** or direction telling it how to arrange, position, interpret etc. the flashable characters.

The Hendrix Composition System (HCS) acts on the same kind of an input information stream. We shall concern ourselves mainly in this manual only with the codes that instruct the program. Collectively they are called "Commands".

2.2 Terminology

Many special words have been used to describe the coding language used to talk to typesetting programs. Definitions seem to cross boundaries, or worse, may have several conflicting meanings. Below is a glossary of several key words. Perhaps, these meanings do not entirely agree with your concepts. Accept what is here for the purpose of understanding this manual with no confusion.

COMMAND is a single or a series of instructions directing the program to control the composition output. HCS uses two types of commands namely:

(1) **Single-keystroke commands** (A key must be incorporated in the terminal keyboard).

(2) **Two-letter function commands** (These use the alphabet only, thus are suited for OCR input as well).

PRECEDENCE CODE used in front of a command (the "start command" code) tells the program that instructions and not text will follow. The "end command"code may also be thought of as a 'precedence code' in the sense that it says

text follows. A "Pi" call is a precedence code that tells the program that the character following it is not normal, but something special. Thus, a precedence code simply gives the program certain conditions for whatever follows.

FUNCTION is the instructional element of a command, and it is a specific, such as; change the point size, quad this line to the left, turn off the flash lamp, etc.

PARAMETER is the numeric value part of a function, the "how much" that is wanted, such as; the "12.5" of twelve and one/half points of leading, the "18" of 18 point size, the "6" of number of columns to be equally divided.

FORMAT is a "memorized string" of commands and/or text characters that can be recalled at will. Remember it is NOT restricted only to commands.

MACRO is similiar to Format above, but is more convenient to use and load. A single key name will recall it.

2.3 Single Key Commands

WHAT THEY ARE: Certain keys on the TTS keyboard layout of the editing terminals have a special significance to the Hendrix Composition Program and are vital to its operation. Others are used so often that they have been included for convenience or to save keyboarding time. This later group has an equivalent 2-letter function, which can be used for OCR input, if no other special codes have been assigned.

BASIC COMMAND KEYS: The keys listed directly below are basic to the command structure of The Hendrix Composition Program. Used properly, they define the syntax of the command language. Improperly used, they will cause an error message, at best; a typographic jumble at worst. Remember them

<	BEGIN	Command	(Specify)	Identifies Two-Letter Function Command Start
>	END	Command	(Specify)	Identifies Two-Letter Funtion Command End
;	SEPARATE	Commands	(Semicolon)	Within a command, divides 'combined' commands
:	SEPARATE	Levels	(Colon)	Within a command, divides multiple levels
,	SEPARATE	Parameters	(Comma)	Within a command, divides parameter values
.	SEPARATE	Picas & Points	(Period)	Within a command, divides picas from points
()	ENCLOSE	General Variable	(Parens)	Within a command, identify General Variable

SECT. II—COMMAND STRUCTURE

2.3 Single Key Commands

SINGLE-KEY OR 2-LETTER COMMANDS?

Below are commands that are normally found on the terminal keyboard but also have equivalent 2-letter function codes. The choice? Single keys surely are the most convenient. But 2-letter functions are necessary for OCR input. This manual uses 2-letter functions when they provide a convenient method of displaying commands in type.

TWO LETTER FUNCTION	SCREEN SYMBOL	KEY NAME	ALT. KEY NAME	COMMAND NAME	COMMAND CODING	TWO LETTER FUNCTION
FB	↑	BOLD	UR	Font Bold	< fb >	**FB**
FL	↓	LITE	LR	Font Lite	< fl >	**FL**
EM	◫	EM	EM SPACE	Em Space	< em >	**EM**
EN	▯	EN	EN SPACE	En Space	< en >	**EN**
TH	‖	THIN	THIN SPACE	Thin Space	< th >	**TH**
JU	J U	JUSTIFY		JUstify line	< ju >	**JU**
QL	←	QUAD LEFT	COPY LEFT	Quad Left	< ql >	**QL**
QC	↔	QUAD CNTR	COPY CNTER	Quad Center	< qc >	**QC**
QR	→	QUAD RIGHT	COPY RIGHT	Quad Right	< qr >	**QR**
QP	¶	QUAD PARA	END PARA	Quad Paragraph	< qp >	**QP**
QM	W X	W/SPACE	QUAD MIDDLE	Quad Middle	< qm >	**QM**
WL	W L	W/....	LDR	Fill With Leaders	< wl >	**WL**
PI	P I	PI		Pi Call	< pi >	**PI**
FM	T ←	MERGE	F M T MERGE	Format Merge	< fm >	**FM**
MM	T →	M. MERGE	MACRO MERGE	Macro Merge	< mm >	**MM**
MC	M C	MACRO C.	MC	Macro Call	< mc >	**MC**
MD	M D	MACRO D.	MD	Macro Define	< md >	**MD**

2.4 2-Letter Function Commands

All 2-letter function commands are designated to the program with a "BEGIN COMMAND"- ❮ followed by the double mnemonic code for the particular function. Some commands then require parameter values to be specified. Some do not. A bullet (•) after the code letters indicates that the command may need parameter values to be valid. See last column for parameter details.
ALL commands finish with an "END COMMAND"-❯ code.

TABLE below groups these commands into categories determined by their usage. The first column shows the two letters in Caps. Either upper or lower case may be used.

Style and Size Control Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
CW•	COLUMN WIDTH	❮**cw** ***p.p***❯
FN•	FONT NUMBER	❮**fn** ***n***❯
PS•	POINT SIZE	❮**ps** ***p.t***❯
VL•	VERTICAL LEADING	❮**vl** ***p.f***❯
CW•	"FOUR-IN-ONE"	❮**cw** ***p.p, n, p.t, p.f***❯
[1]**FB**	FONT BOLD (or Italic)	❮**fb**❯
[1]**FL**	FONT LIGHT	❮**fl**❯
RB	RAIL BOLD	❮**rb**❯
RI	RAIL ITALIC	❮**ri**❯
WB	WUN-LINE BOLD (or Italic)	❮**wb**❯
WF•	WUN-LINE FONT Change	❮**wf** ***n***❯
WZ	WUN-LINE ZERO Leading	❮**wz**❯
AW•	ALTER WIDTH	❮**aw** ***u or -u***❯
PW•	POINT WIDTH	❮**pw** ***p.t***❯

Spacing Control Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
VA•	VERTICAL ADVANCE Leading	❮**va** ***p.f or -p.f***[2] ❯
VB•	VERTICAL BACK Leading	❮ **vb** ***p.f***[2] ❯
VE•	VERTICAL EXTRA Leading	❮**ve** ***p.f or -p.f***[3] ❯
VU•	VERTICAL UNIT Leading	❮**vu** ***u or -u***[2] ❯
VM	VERTICAL MARK	❮**vm**[2] ❯
VR	VERTICAL RETURN to Mark	❮**vr**[2] ❯
VP	VERTICAL POINT	❮**vp**[2] ❯
VC	VERTICAL CLEAR Leading Counter	❮**vc**[2] ❯
HP•	HORIZONTAL Space in PICAS	❮**hp** ***p.p***❯
HU•	HORIZONTAL Space in UNITS	❮**hu** ***u or -u*** ❯
[1]**EM**	EM Fixed Space	❮**em**❯
[1]**EN**	EN Fixed Space	❮**en**❯
[1]**TH**	THIN Fixed Space	❮**th**❯

[1] This command has a more convenient single-key form. *(See previous table on page 2-3.)*
[2] The command or parameter portion marked requires a reverse leading capability.
[3] Reverse lead amount limited to current line leading value.

SECT. II—COMMAND STRUCTURE

2.4 2-Letter Function Commands

Hyphenation and Justification Control Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
DH•	DEFINE HYPHENATION Mode	⟨dh*c, h*⟩
AH	ALLOW HYPHENATION Mode	⟨ah⟩
XH	CANCEL HYPHENATION Mode	⟨xh⟩
AL	ALLOW LETTERSPACE Mode	⟨al⟩
XL	CANCEL LETTERSPACE Mode	⟨xl⟩
AD	ALLOW DICTIONARY Mode	⟨ad⟩
XD	CANCEL DICTIONARY Mode	⟨xd⟩
HO	HOMOGRAPH (Discret. Hyphen)	⟨ho⟩
AN	ALTER MINIMUM Spaceband	⟨an*u*⟩
XN	CANCEL MINIMUM Spaceband	⟨xn⟩
AM	ALTER MAXIMUM Spaceband	⟨am*u*⟩
XM	CANCEL MAXIMUM Spaceband	⟨xm⟩
RL	RAGGED LEFT	⟨rl⟩
RR	RAGGED RIGHT	⟨rr⟩
RC	RAGGED CENTER	⟨rc⟩
RX	RAGGED CANCEL	⟨rx⟩

Line Control Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
[1]**JU**	JUSTIFY Line	⟨ju⟩
[1]**QL**	QUAD LEFT	⟨ql⟩
[1]**QC**	QUAD CENTER	⟨qc⟩
QR	QUAD RIGHT	⟨qr⟩
[1]**QP**	QUAD PARAGRAPH	⟨qp⟩
[1]**QM**	QUAD MIDDLE	⟨qm⟩
[1]**WL**	Fill WITH LEADERS	⟨wl⟩
WR	Fill WITH RULES	⟨wr⟩
WX•	Fill WITH Character "X"	⟨wx⟩ *X*

Indent Control Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
IT•	INDENT TAKE	⟨it*l.l, r.r*[6]⟩
IX	INDENT TAKE CANCEL	⟨ix⟩
HI•	HANG INDENT	⟨hi*l.l, r.r, w, d*[6]⟩
HH•	HANG HERE	⟨hh*d*[6]⟩
HX	HANG CANCEL	⟨hx⟩
HR	HANG REPEAT Indent	⟨hr⟩
IH	INDENT HANG	⟨ih⟩
IR	INDENT Hang RESET	⟨ir⟩
IP•	INDENT PARAGRAPH	⟨ip*l.l, r.r*[6]⟩

[1] This command has a more convenient single-key form. *(See previous table on page 2-3.)*
[6] One or more parameter values may be omitted.

2.4 2-Letter Function Commands

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
AA	ALLOW AUTO Paragraph Indent	⟨aa⟩
XA	CANCEL AUTO Paragraph Indent	⟨xa⟩
LL•	LENGTH LEFT Fit	⟨ll⟩
LX	LENGTH LEFT CANCEL	⟨lx⟩
LR•	LENGTH RIGHT Fit	⟨lr*r.r, d.d:...*⟩
RX	LENGTH RIGHT CANCEL	⟨rx⟩
SL•	SKEW LEFT Indent	⟨sl*l.l, d.d:...*⟩
SR•	SKEW RIGHT Indent	⟨sr*r.r, d.d:...*⟩
SX	SKEW CANCEL	⟨sx⟩

Tabbing Command Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
TA	TAB ACTIVATE Mode	⟨ta⟩
TX	TAB CANCEL Mode	⟨tx⟩
TK	TAB KILL Remaining Columns	⟨tk⟩
TS•	TAB STOPS	⟨ts*i,s1,s2,s3,..*[6]⟩
TW•	TAB WIDTHS	⟨tw*i,w1,w2,w3,..*[6]⟩
TE•	TAB EQUAL Number of Columns	⟨te*i,n*[6]⟩
TP•	TAB PROPORTIONAL Each Column	⟨te*i,r1,r2,r3,..*[6]⟩
TT•	TAB TEXT Here	⟨tt⟩
TI•	TAB INDENT Each Column	⟨ti*l.l,r.r*[6]⟩
TV•	TAB VARIABLE Space	⟨tv⟩

Compose-For-Display Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
ST	STRIP TAKE Compose	⟨st⟩
AE	ALLOW ESTIMATOR for Heads	⟨ae⟩
XE	CANCEL ESTIMATOR for Heads	⟨xe⟩
NC	NO COUNT Line	⟨nc⟩
GD•	GALLEY DEFINE	⟨gd*w, d, g, n, e,*[6] ⟩
GE	GALLEY END Point	⟨ge⟩
GF	GALLEY FREE (Discretionary End)	⟨gf⟩
PC•	PHOTO COMPOSE Device and Font Select	⟨pc*n*⟩

Miscellaneous Command Group

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
PI	PI Character (and Font) Call	⟨pi⟩ *'x'*
XF	CANCEL FLASH	⟨xf⟩
AF	ALLOW FLASH	⟨af⟩
VI	VISUALS START on Tape	⟨vi⟩
XV	VISUALS END on Tape	⟨xv⟩

[6] One or more parameter values may be omitted.

SECT. III—COMPOSITION COMMANDS

2.4 2-Letter Function Commands

Special Typesetter Model Commands

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
CG Videosetter UNIVERSAL		
NV	NEGATIVE VIDEO	⟨nv⟩
PV	POSITIVE VIDEO	⟨pv⟩
NR•	NORMAL Width ROMAN	⟨nr*g*[4]⟩
NI•	NORMAL Width ITALIC	⟨ni*g*[4]⟩
NB•	NORMAL Width BOLD	⟨nb*g*[4]⟩
ND•	NORMAL Width BOLD ITALIC	⟨nd*g*[4]⟩
CR•	CONDENSED Width ROMAN	⟨nb*g*[4]⟩
CI•	CONDENSED Width ITALIC	⟨ci*g*[4]⟩
CB•	CONDENSED Width BOLD	⟨cb*g*[4]⟩
CD•	CONDENSED Width BOLD ITALIC	⟨cd*g*[4]⟩
ER•	EXPANDED Width ROMAN	⟨er*g*[4]⟩
EI•	EXPANDED Width ITALIC	⟨ei*g*[4]⟩
EB•	EXPANDED Width BOLD	⟨eb*g*[4]⟩
ED•	EXPANDED Width BOLD ITALIC	⟨ed*g*[4]⟩
Autologic APS-4		
SO•	SET OBLIQUE	⟨so*a*⟩
XO	CANCEL OBLIQUE	⟨xo⟩
Dymo PACESETTER		
IE	INITIALIZE EIGHT TABLE	⟨ie⟩
IN	INITIALIZE NINE TABLE	⟨in⟩

Formats and Macros

TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
FD•	FORMAT DEFINE	⟨fd *f.name*⟩
FE	FORMAT END	⟨fe⟩
[5]MD	MACRO DEFINE	⟨md*n*⟩
[5]MC	MACRO CALL	⟨mc*n*⟩
[5]FM	FORMAT MERGE	⟨fm⟩
[5]MM	MACRO MERGE	⟨mm⟩
SB•	STRING BEGIN	⟨sb*n*⟩
SE	STRING END	⟨se⟩
SC	STRING CANCEL	⟨sc⟩
GV	GENERAL VARIABLE DEFINE	⟨gv*a:v1;v2:v3: ...*⟩
(A)	GENERAL VARIABLE CALL	(a) to (z)

[4] The parameter value may be omitted.
[5] Use more convenient single-key form. This command available only for OCR input. *(See table on page 2-3.)*

Only the codes listed in this table are valid. No other symbol, spaceband, carriage return, elevate, etc. can be inserted within the command unless it is a legal parameter, or a separator. Any violations will cause an error condition. The commands will be ignored and an error message displayed at the end of the story.

2.5 Elements of a 6400 Command

The Hendrix Composition System (HCS) uses a simple language. It breaks down into "words" (commands) and "sentences" (formats). It has "adjectives and adverbs" (macros and general variables). There even is a grammar. All this will be explained in this manual. If you make an error in the syntax (or use illegal parameters for your system), then the program can detect these mistakes. When it does, it issues a precise error message telling you just where you erred.

BASIC COMMAND: A basic command is composed of four parts arranged in the following pattern.

(BEGIN) (FUNCTION) (PARAMETER) (END)

In actual practice, it would take some such form as this.

〈 (Point Size) (Five and a Half) 〉

Finally, the command itself would appear in this fashion.

〈 ps 5.5 〉

Several commands may be issued in succession or they may be in a combined form as a single command. This eliminates several keystrokes, an important consideration when trying to fit a long format into the fixed character area available.

If any command vital to typesetting is omitted at the start of a job, the program supplies a *"default value"* which is used instead. These *"default values"* were chosen as part of the "specification package" at the time of installation. Consult your Computer System Operator for actual numbers.

2.6 Separators

Commands or parameters may be combined in several ways. In order for the program to keep track, you must adhere to specific rules.

PERIOD (.) - separates PICAS from POINTS
POINTS from FRACTIONS

In the HCS program a valuable feature is the ability to specify picas only, points only, or both. When a parameter value has

2.6 Separators

a whole and a fractional part, separate these parts by a decimal (ie., 12½ becomes 12.5). Any zeros which precede a number, on either side of the decimal point, need not be used. For instance, 12 picas, 6 points may be either 12.06 or 12.6.

If the parameter is a whole value only, then the decimal point is not needed. Thus, 9 picas can be written as merely a 9. The use of a 9.0 or 09 is not necessary, but it is not invalid and will be accepted by the program. Here are some further examples.

NOTE: At this point, it's not important to know the meaning of each command.

〈cw 10〉	(10 picas)	10 pica Col Width
〈cw 8.24〉	(8 picas + 24 pts)	10 pica Col Width
〈cw .120〉	(120 pts)	10 pica Col Width
〈cw 10.6〉	(10 picas, 6 points)	10½ pica Col Width
〈vl 10〉	Vertical leading 10 points	
〈vl 10.5〉	Vertical leading of 10½ points	

COMMA (,) - separates PARAMETERS

If several parameters are possible, within a command, then these parameters are separated by commas. If a parameter is not necessary, the numerical value can be omitted but NOT the commas. The only exception is the last value. Examples below show each of these points.

BEFORE **〈cw22.9〉 〈fn3〉 〈ps9〉 〈vl12〉**
AFTER **〈cw22.9 , 3 , 9 , 10〉**

CMMD	**〈hi1 , 2 , 3 , 4〉**	**All parameter values present**
OR	**〈hi , 2 , 3 , 4〉**	**First parameter value missing.**
OR	**〈i1 , , 3 , 4〉**	**Second parameter value missing.**
OR	**〈i1 , 2 , , 4〉**	**Third parameter value missing.**
OR	**〈i1 , 2 , 3 〉**	**Fourth parameter value missing. (NO COMMA NEEDED).**
OR	**〈hi1 , , , 4〉**	**Second and Third parameter value missing.**

SEMICOLON (;) - separates MULTIPLE COMMANDS

If several commands are to be entered one after the other, the semicolon can be used to cut keystrokes. Like this:

BEFORE **〈xx〉 〈ah〉 〈ae〉 〈cw22〉 〈nr2,10〉 〈it4〉**

AFTER **〈xx : ah ; ae ; cw22 ; nr2,10 ; it4〉**

COLON (:) - separates MULTIPLE LEVEL PARAMETERS
Sometimes, a command allows several "groups" of parameter values, so as to permit uential actions. These groups (called "levels") are separated from one another by the use of colons (:).

SIX SKEW RIGHT'S **‹SR 1,3 : 2,4 : 3,-5 : 4.6,3.3 : 1,7 : 4.6,2.9 ›**

2.7 Error Messages

If there is an error of any kind in the composition commands, the program will detect them.Errors could be syntax because the command rules were not followed exactly. Information may be missing which is vital to the command. The program may not be able to carry out the logic that is requested. The error will be noted internally in the program, the offending command is passed over and processing is continued. At the end of the job a list of errors will be given. The error messages might appear something like these shown below. The word "ERROR" is first printed. The second column gives the line number on which the error was encountered. The last column prints out that portion of the format or copy where the mistake was first detected. The NOTES below do not appear, they are explanatory only.

ERROR	00007	:hi3,2,,4›
		NOTE—The : should have been a ;
	00024	ation Command ‹nb2;A4;fe›
		NOTE—This part of a format exceeded the capacity as stated in the message below
	00024	FMT OVER 124 CHARS
	00034	‹cw10;it12›
		NOTE—The command above asks for 12 pica indent on 10 pica col
	0024	NEG COL WIDTH

SECTION III
COMPOSITION COMMANDS

3.1 Introduction

This section comprises the basis of the 6400 HCS program. It's the *only* language that the computer understands! To master mark-up, you must first learn that language. Learn its codes, its commands, its capabilities, its restrictions, its limits and also, its advantages and power. That's what this section is about.

In most instances, each page has been individualized and contains either a single composition command or closely related commands. Instructions, explanations, examples; they are all on the same page. Furthermore, the pages are designed to give detailed information, yet simultaneously they emphasize the important elements so as to serve as quick-reference pages.

3.2 Using Command Pages

On the back of this leaf (and facing the first command page) is a sample page explaining each element.

AB• *HAS PARAMETER VALUES*

CANCEL WITH XY *HAS CANCEL COMMAND*

Two-Letter function name

X.X Command Group

EXAMPLES of command AB

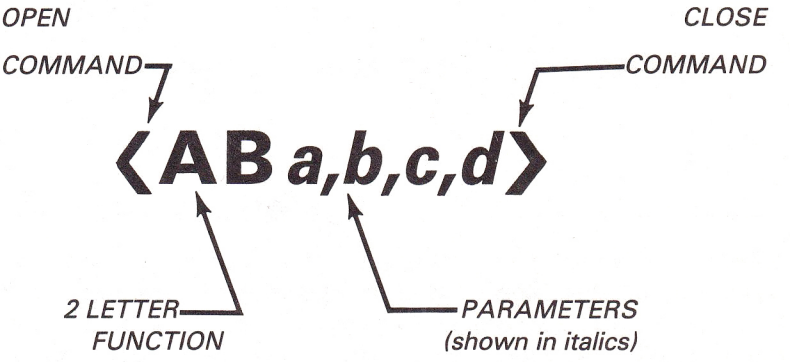
Using the command above, comments about the example will be given in this italic style of type so as to be distinctive from the actual sample below:

This is the actual example. Whenever possible the basic typesetting is in the style of a newspaper story column, namely; Corona typestyle, 9 point size on 9 and a half leading, with a column width of 11 picas.

These examples limit themselves to illustrating and clarifying the actions of the command on this page only. Pitfalls, peculiarities, unusual situations are included if space is available.

NAME OF COMMAND

(Alternate names given here)



COMMANDS are described and their actions explained in this paragraph of the page. Most pages have a single command. To make it easy to find a particular command the 2-letter mnemonic code is put in large type at the top outer area. If it is followed by a bullet (•) look for parameters, which are sometimes optional, sometimes mandatory. If there is no bullet, the command is functional only and needs no values. The command itself is shown in pictorial form above this paragraph. Parameter names and units (including important limitations) are flagged with arrows for quick reference. Examples of the command in use are shown in the side columns.

NOTE: Information is shown in smaller type, like this, if it is (1) of secondary nature. (2) an exception or a special case. (3) an explanation too specific for the main text.

BEGINS will state the conditions when the command starts to function.

ENDS will tell when the command state or action will cease. If there is a corrsponding CANCEL command it will appear in large type at the top of the page, in addition to its mention here.

EXCEPT generally lists those situations which will cause an error message to appear at the end of the job file. Alternately, limitations are quoted here if the program rejects values that exceed such limits.

NOTE: Limits are due to typesetter ranges or arbitrary limit values for your particular installation.

SECT. III—COMPOSITION COMMANDS

3.3 Style and Size Group

CW•
Column Width

COLUMN WIDTH Change

(Also called "Line Length, Measure", "Page Column Size")

Picas and/or Points

⟨cw *p.p*⟩

COMMAND establishes or changes the length of the line put out by the phototypesetter. Usually, this is called the column width or measure. The value is specified in either picas alone, picas and points or total points alone. If points *only* are used they must be preceeded by a period. The maximum length allowed depends upon the capacity of the particular model of typesetting device. Fill in the space provided at the bottom of this page when you know the correct amount.

BEGINS on the current line, but *only* if issued at the beginning, before any copy. When it is given in the middle of a line, the new column width does not appear until the *Next Line.*

ENDS at the next valid COLUMN WIDTH **⟨cw⟩** command. At times, it may appear to have been changed because of indents on the right, run-arounds, notches, etc., but the program *always* remembers the current column width, unless it is intentionally reissued.

EXCEPT the command is ignored, and error messages are flashed:

1. If the total value of points *exceeds* 1000, (ie, cw.1000).
2. If the width asked for is greater than that of the photocomposition machine itself.
3. If there is no period between picas and points.

_____ _____ Maximum line length—in picas

_____ _____ Maximum line length—in points

TYPESETTER 1 TYPESETTER 2

EXAMPLES of command PS

Set column to a width of ten picas and 9 points. Give command before first line.

⟨cw10.9⟩
The paragraph you are now reading shows how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Change the length of the line to make it 8 picas even

⟨cw8⟩
The paragraph you are now reading demonstrates how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates and then issues the necessary instructions to the typesetter.

Now let's insert the column width command in the middle of a line to show that the action will not take place until the NEXT LINE.

The paragraph you are now reading demonstrates how the **⟨cw7⟩** Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then it issues the instructions.

Column widths may be specified using ONLY POINTS. For instance, 120 points is equivalent to 10 picas. Remember, the PERIOD must be BEFORE the total point value, thus:

⟨cw.120⟩
The paragraph you are now reading demonstrates how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates and then issues the necessary instructions to the output typesetter.

FN•

Font Number

EXAMPLES of command FN

Issue each line of type in a different font style.

‹fn2› This is Font No. 2
‹fn3› This is Font No. 3
‹fn4› *This is Font No. 4*
‹fn13› This is Font No. 13
‹fn14› *This is Font No. 14*
‹fn27› This is Font No. 27
‹fn30› *This is Font No. 30*
‹fn65› **This is Font No. 65**
‹fn72› ***This is Font No. 72***

Change fonts within the lines of type.

‹fn1› Demonstrating changes of typeface. ‹fn2› *Demonstrating changes of typeface.* ‹fn3› Demonstrating changes of typeface. ‹fn4› *Demonstrating changes of typeface.* ‹fn8› *Demonstrating changes of typeface.* ‹fn11› Demonstrating changes of typeface. ‹fn48› *Demonstrating changes of typeface.* ‹fn63› **Demonstrating changes of typeface.**

FONT NUMBER Change

(Also called "Face No.", "Style No.", "Typeface No.")

Number given to font

‹fn*n*›

COMMAND selects a new font style from the array possible within the photocomposing device. Specific arrangements vary greatly between models and manufacturers, especially between optically-projected characters (such as the Dymo Pacesetter or Compugraphic Unisetter) and CRT or Laser generation, which are capable of *"pseudo"* fonts styles (such as Compugraphic Videosetter, APS-5 and Dymo Laser DCL-10). Generally speaking, fonts come in groups of 8, 12 or 16. For instance:

Disc A contains Fonts 1-16
Disc B contains Fonts 17-32
Disc C contains Fonts 33-48
etc.

Grid 1 contains Fonts 1-12
Grid 2 contains Fonts 13-24
Grid 3 contains Fonts 25-36
etc.

Strip A contains Fonts 1-8
Strip B contains Fonts 9-16
Strip C contains Fonts 17-24
etc.

(See Section 3-11 for details and special commands for your particular output machine.)

BEGINS immediately, any place within the line. There is no limit to the number of font changes permitted in the same line.

ENDS at next occurence of valid FONT NUMBER ‹fn› command.

EXCEPT

1. If no font is specified, the *"default"* font will be used.
2. If a non-existent font is asked for, an error message will come up on the screen.
3. If a non-existent point size is next requested, it will inhibit setting.

PS•
Point Size

SECT. III—COMPOSITION COMMANDS
3.3 Style and Size Group

POINT SIZE Change

(Also called "Type Size, Type Height, Set Height")

Points and Tenths

⟨ps *p.t*⟩

COMMAND selects sizes of type valid for the output device. The point size of a type is the actual physical space it occupies on paper. Changes in point sizes are accomplished by changing lenses in optical typesetters. In CRT and Laser machines, increasing or decreasing the vertical travel of the light beam does the trick. Once set-up, however, a given point size is *always* the same height, for a given font.

BEGINS immediately, anywhere within a line. There is no limit to the number of size changes within a line.

ENDS with the next valid POINT SIZE **⟨ps⟩** command. Note that each time the point size is changed, the program automatically re-establishes the standard table values of:

1. Minimum and Maximum Spaceband Units.
2. Vertical Leading equal to point size (if not requested) *(See Automatic Leading on next page).*
3. Condensing Value *(CRT only).*
4. Expansion Value *(CRT only).*

EXCEPT

1. If no point size is specified the *"Default"* size contained in "HSETUP" will be used instead.
2. If a non-existent point size is requested, an error message will be output and the offending line of copy flagged.
3. If fractional point sizes are available, use the exact amount (in tenths). For instance, for 5 and one-half use 5.5 as the parameter value.

_____ _____ Pt Size Increments..FROM ____pts TO ____pts.

_____ _____ Pt Size Increments...FROM ____pts TO____pts.

TYPESETTER 1 TYPESETTER 2

EXAMPLES of command PS

Set a paragraph in 8 point type. DO NOT specify any leading.

⟨ps8⟩ The paragraph you are now reading demonstrates how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates and then issues the necessary instructions to the output typesetter.

Issue each line of type in a different point size. Note that the leading is automatically adjusted to the point size. See next page "Automatic Leading".

⟨ps7⟩ This is a line of 7 point.
⟨ps9⟩ This is a line of 9 point.
⟨ps14⟩ This is 14 point.
⟨ps18⟩ This is 18 pt.
⟨ps24⟩ 24 point.
⟨ps36⟩ 36 pt.

*Mix several sizes in succession. Use enough copy to make several lines, then quad left. Note that the **FIRST** size in the line now establishes the leading.*

14pt, 7pt, 24pt, 5.5pt, 11pt, 9pt, 16pt, 28pt, 5.5 point, 12pt, 14pt, 11pt, 18pt, 6 pt

VL•

Vertical Lead

EXAMPLES of command VL

First let's look at a block of 8-point copy set solid.

〈ps 8〉 〈v l8〉
The paragraph you are now reading shows how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Now increase it to 8½ leading. Not much difference, just noticeable.

〈vl 8.5〉
The paragraph you are now reading shows how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Next go to 10 pt leading. It's a big difference.

〈vl10〉
The paragraph you are now reading shows how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

The same copy in CAPS can be set on "negative' leading. For instance 8/7.

〈ps 8〉 〈vl 7〉
THE PARAGRAPH YOU ARE NOW READING SHOWS HOW THE HENDRIX 6400 SYSTEM ACCEPTS A COMPOSITION COMMAND, INTERPRETS IT, CALCULATES, AND THEN ISSUES THE NECESSARY INSTRUCTIONS TO THE OUTPUT TYPSETTER.

VERTICAL LEADING

(Also called "Vertical Space", "Film Advance", "Inter-line Spacing")

Points and Fractions

〈vl *p.f* 〉

COMMAND specifies the amount of vertical space, measured from baseline-to-baseline. In most cases, this means the distance traveled by the photographic material (the film advance).

Leading a fractional part of one point depends upon the ability of the output typesetter.

Tenth-point machines—Use .1 to .9 (or nothing)
Quarter-point machines—Use .25, .5, .75 (or nothing)
Half-point machines—Use .5 (or nothing)

The vertical leading value is stored in the program and is used to *first advance* the paper each time, before a line is flashed.

Vertical leading always includes the point size of the type **plus** any "extra" space, which may be *more than* the point size (ie. 10 on 12 - "leaded type") or may be *less than* (ie. 36 on 30 - "negative leading"). If the leading is equal to the point size, the text has been "set solid". Some photocomposition machines advance paper *before* the line is flashed, others *after*. Regardless, the program always adds extra leading **above** the current line, in a manner known as "top leading."

AUTOMATIC LEADING takes place if a point size is given and it is **not** immediately followed by a leading value. In such a case, the program automatically issues a leading value which is equal to the point size (ie. " set solid"). If there are several sizes mixed on the line, the **first** point size is used. *(See samples of point size changing on previous page).*

BEGINS at the start of each line, but can be issued anywhere within the line. Remember, **first** the paper advances, **then** the line is set. Extra new leading will appear above the current line.

ENDS at the next valid VERTICAL LEAD **〈vl〉** command.

TYPESETTER 1	TYPESETTER 2	Units used for fractions of leading.

SECT. III—COMPOSITION COMMANDS

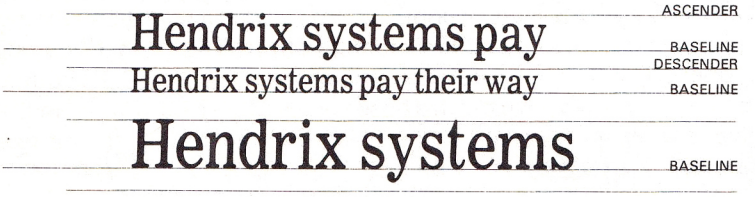
3.3 Style and Size Control Group — Vertical Lead

(VL) continued

EXCEPT

1. If no vertical leading command is issued, the "default" value in the "HSETUP" table will be used, unless a point size is given, in which case "automatic leading" takes place.
2. If the period is omitted between the points and fractions of a point in the parameter value, an error message will show up.
3. If the parameter value exceeds 1000 points, i.e. **‹vl1000›** , an error message comes up.

LEADING is the most difficult of the functions so far presented. It is hard to visualize or anticipate the correct values during layout or composition. This is especially true when changing point sizes from line to line. The illustrations and table below may clarify or expedite such leading commands.



In the example above, the Vertical Leading is shown by the vertical arrow line. Broadly speaking, capital letters use 2/3 of the point size body (Ascender-to-Baseline). The remaining 1/3 hangs down below the base to make room for the descender (Baseline-to-Descender).

FAST FORMULA for the tightest fit between any two lines of differing point sizes is determined thus:

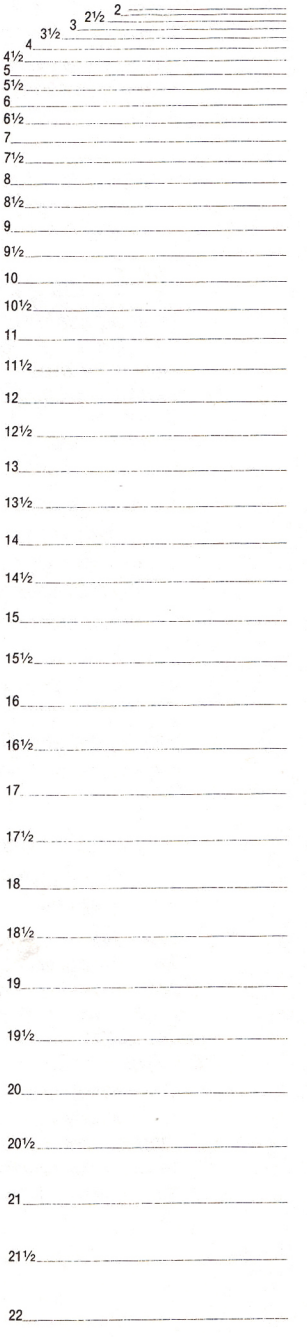
1/3 of PRESENT line + 2/3 of NEXT is **MINUMUM LEADING**

Study the examples above, then refer to the chart below. Use it in the future for quick reference when changing point sizes.

POINT SIZE (FROM)	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**14**	**18**	**24**	**30**	**36**	**42**	**48**	**54**	**60**	**72**
1/3 Point Size	2	2	3	3	3	3	4	4	5	6	8	10	12	14	16	18	20	24
PLUS																		
2/3 Point Size	3	4	4	5	6	7	7	8	9	12	16	20	24	28	32	36	40	48
POINT SIZE (TO)	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**14**	**18**	**24**	**30**	**36**	**42**	**48**	**54**	**60**	**72**

EXAMPLES of command Leading Ruler

The scale below can be used to determine the vertical leading between two lines within 1/2 point. It shows how precise leading can be.



CWoooo

"Four-in One" Command

EXAMPLES of command CW

Let's set-up a typical newspaper column using
Font 1, 9/10 pt x 10½ col

‹cw10.6, 1, 9, 10›
The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Without disturbing column width (by starting with FN) change to 8/9 Italics

‹fn2, 8, 9›
The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Now we go to a 6 point classified ad type set-up.
(Ad Font 49, 6 pt, 9½ col)
DO NOT specify Vertical Lead.
(Automatic Leading will supply the value)

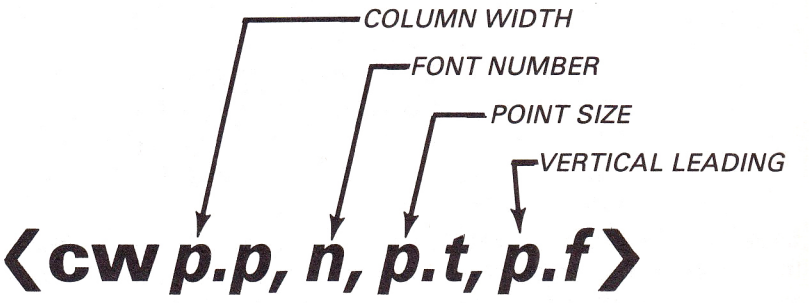
‹cw9.6, 49, 6›
The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Change col width and font ONLY. Just omit values for size and leading.

‹cw7,16›
The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

"FOUR-IN-ONE" Change

(Also called "Combination Command")



COMMANDS may be combined to provide a shorthand method tha reduces markup time and keyboarding effort. The "4-in-1" combinatior is opened, individual parameter values are separated by commas, anc then the command is closed. These values MUST be entered in the orde given. If all of them are not needed omit according to the example below:

‹cw10.3,6,9,10›	Full Command.
‹cw10.3,6,,10›	Omit Point Size
‹cw10.3,6›	Omit Size and Lead (Note: Last comma dropped).

In addition, a sequence may be started somewhere in the middle by using the proper 2-letter function code, thus:

Full Combination	**‹cw10.3,6,9,10›**
Start with Font Number	**‹fn6,9,10›**
Start with Point Size	**‹ps9,10›**

BEGINS, ENDS & EXCEPT conditions that apply to each individual function, as explained in prior commands, of course also apply here when they are combined.

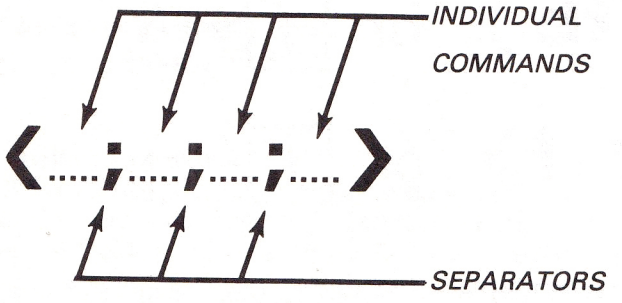
SECT. III—COMPOSITION COMMANDS

3.3 Style and Size Control Group

‹ ..... ; ..... ›
Combined Command

COMBINED COMMAND

(Also called "Combination Command")



ALL COMMANDS can be combined or run together by using a semicolon to separate each from the other. For example, the four commands that have been presented so far can be combined in any order, like this:

INDIVIDUAL COMMANDS

‹cw20› ‹fn3› ‹ps24› ‹vl30›

COMBINED COMMANDS

‹cw20 ; fn3 ; ps24 ; vl30›

or

‹ps24 ; vl30 ; fn3 ; cw20›

or

‹fn3 ; vl30 ; cw20 ; ps24›

ADVANTAGES of combining commands are:

1. Less keyboarding.
2. More familiar key symbol.
3. Easier to read, print and interpret.
4. Saves space in formats and macros.

COMBINED COMMANDS

The following is typical of a lengthy combined command. (A step-by-step explanation follows.)

‹xx; dh*5,4*; ah; xl; an*4*; aa; cw *13,1,9,10*; it *1,1*; te*2,3*; vm›

‹xx; Cancel All indents/tabs, in case any remain from previous commands.

dh*5,4* Define Hyphenation routine, ignore words of 5 chars or less, hyphenated lines-in-a-row limited to 4.

ah; Allow Hyphenation mode,

xl; but Cancel Letterspace so that all justification takes place between words only,

an*4;* using a Minimum Spaceband of 4 units.

aa; Allow Automatic paragraph indentation based on an internal table of values.

cw *13,1,9,10;* Four-in-one command (see opposite page).

it *1,1;* Indent Take of 1 pica left, 1 pica right.

te*2,3;* Tab Equal **set-up only** (2 pica indent, then divide space remaining into 3 equal columns).

vm› Finally, place a Vertical Mark as soon as copy starts.

FB/FL

Font Bold/Font Light

EXAMPLES of command FB, FL

One of the most commonly used keys on newspaper work is the Bold/Light key. Below is shown a sample paragraph with these commands visible. Each of the examples then exhibits the companion bold that is called out. These "pairings" have been set up in advance in your program.

COPY
The paragraph you are now reading demonstrates how the Hendrix 6400 accepts a **⟨fb⟩** Composition Command, **⟨fl⟩** interprets it, calculates and then issues the necessary instructions to the output typesetter.

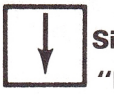
The paragraph you are now reading demonstrates how the Hendrix 6400 accepts a **Composition Command,** interprets it, calculates and then issues the necessary instructions to the output typesetter.

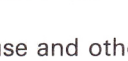
The paragraph you are now reading demonstrates how the Hendrix 6400 accepts a **Composition Command,** interprets it, calculates and then issues the necessary instructions to the output typesetter.

The paragraph you are now reading demonstrates how the Hendrix 6400 accepts a **Composition Command,** interprets it, calculates and then issues the necessary instructions to the output typesetter.

FONT BOLD (or Italic)/FONT LIGHT

⟨fb⟩ or **Single-Key Called "BOLD"**

⟨fl⟩ or  **Single-Key Called "LIGHT"**

COMMAND(S) select the companion "Bold" and/or a return t Light, per tables. The concept is the same as Upper Rail/Lower Rai There is a corresponding single key, on the terminal which appears o screen as  This is the easiest way to switch back and forth.

For OCR use and other cases, the full command FONT BOLD **⟨fb⟩** c FONT LITE **⟨fl⟩** is also provided.

Any two fonts may be so "paired", which is to say may be alternate with a single key stroke. Paired fonts which share the same width assign ments letter for letter, are "duplexed' as in a hot metal metrix. Whether Roman style is "paired" with a Bold, or "paired" with an Italic, depend on the next command on the following page. The option is yours.

BEGINS immediately anywhere within the line.

ENDS at the next font change.

RB/RI
Rail Bold/Rail Italic

RAIL BOLD/RAIL ITALIC

‹rb› ‹ri›

COMMAND is used to set up either a Bold, or an Italic as the "companion" font when the FONT BOLD **‹fb›** command is used. Another way of looking at it: the Upper Rail becomes Bold when "Rail Bold" **‹rb›** is keyed first. Upper Rail is redefined as Italic when "Rail Italic" **‹ri›** is issued. The Rail Bold/Rail Italic commands do not call out fonts, they merely define the font that Font Bold secures.

BEGINS immediately in the sense that the condition of the **‹fb›** command has been changed.

ENDS at the next **‹rb›** or **‹ri›** as the case may be and also, at the end of a job, since the start of a job automatially re-establishes Font Bold.

EXCEPT

1. Rail Bold **‹rb›** condition is automatically in effect at the start of each job.

EXAMPLES of command RB, RI

To start off, note the effect of **‹fb›** *and* **‹fl›** *in the paragraph below.*

The paragraph you are now reading demonstrates how the **‹fb›** Hendrix 6400 **‹fl›** System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the typesetter.

By using the **‹ri›** *command we redefine Font "Bold" as Font "Italic", thus;*

‹ri›
The paragraph you are reading demonstrates how the **‹fb›** *Hendrix 6400* **‹fl›** System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the typstter.

We can re-establish the "Bold" in Font Bold with a **‹rb›** *command, in this manner.*

‹rb›
The paragraph you are now reading demonstrates how the Hendrix **‹fb›** Hendrix 6400 **‹fl›** System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the typesetter.

WB/WF•

Wun-line Bold/Wun-line Font

EXAMPLES of command WB, WF

Many newspapers still use a classified ad style with the first line of the ad in Bold. It was very popular when the Linotype was prevalent because it was automatic. (At the end of each line you "came off the rail".) The Wun-line Bold *makes it easy to duplicate this effect.*

〈wb〉 VENDING ATTENDANT, Male or female, 1.p.m.-9 p.m. Waltham area. We will train. Call 890-3256

〈wb〉 EXPERIENCED waste paper sorter wanted. Apply in person,. 98 Taylor St., Dorchester, Mass.

〈wb〉 HOLIDAY INN of Framingham is looking for restaurant help. Please apply in person, 11a.m. to 5 p.m.

Take an established ad, like the one shown below:

COPY and AD:
〈fn9,18〉 Boston Globe
Classified
〈fn91,36〉 929-1500

Boston Globe
Classified
929-1500

When inserting an additional line in an ad the Wun-line Font *lets you specify a typestyle with no regard to what comes before or after.*

COPY and AD:
Boston Globe
〈wf87〉 CALL
Classified
929-1500

Boston Globe
CALL
Classified
929-1500

WUN-LINE BOLD (or Italic)
WUN-LINE FONT Change

〈wb〉

〈wf *n*〉

NUMBER of font

COMMANDS known as "Wun-line" make a change for a single line only. At the beginning of the next line, the program automatically returns to the values that were present before the "Wun-line" command.

WUN-LINE BOLD **〈wb〉** command acts very much like Font Bold *(see 2 pages prior)* except that it is temporary and acts for only a single line. Note that it is also "preset" to a Bold or Italic by the Rail Bold/Rail Italic. *(This is explained one page back).*

WUN-LINE FONT **〈wf〉** command is in turn similiar to the Wun-line Bold but is used to bring forth any available style. It is especially useful for copy that is to be merged with a format. By using a **〈wf〉** command a specific style can be requested for a particular line, with assurance that the copy will return to the original style on the next line, even if that style be unknown to the operator. This usually is the case when calling a format or using a merge.

BEGINS anywhere within a line, but is "self cancelling" at the end of the line.

ENDS at the termination of line in which it is issued. Font returns to that used prior to the Wun-line Bold or Wun-Line Font command.

EXCEPT this command will be ignored:

1. If the font selected is already in use.
2. If the font selected is not in the customer's table.

WZ

Wun-line Zero lead

WUN-LINE ZERO Leading

(Also called "Zero Lead")

⟨wz⟩

COMMAND will inhibit any film advance for one line only. This is a helpful command if one line must "go on top of another".

BEGINS at the beginning of a line. This command MUST be issued before any text on the line.

ENDS automatically after the line has been completed.

EXAMPLES of command WZ

This is a very useful command. For sports line-ups it's nice to be able to Quad Left and Quad Center and Quad Right all on the SAME line! Wun-line Zero Lead gives us this ability. Here's how. Alternately Quad Middle and Quad Center, but use **⟨wz⟩** *in-between lines.*

COPY
Barnes **⟨qm⟩** Simpson **⟨ju⟩**
⟨wz⟩ Center **⟨qc⟩**
Henderson **⟨qm⟩** Monroe **⟨ju⟩**
⟨wz⟩ Right Guard **⟨qc⟩**

Examples:

Barnes	Center	Simpson
Henderson	Right Guard	Monroe
Gorman	Left Guard	Borowiak
Weiss	Right Forward	Williamson
Anderson	Left Forward	Lee

AW•

Alter Width

CANCEL WITH
XW

ALTER WIDTH of Font

(Also called "White Space Reduction")

Units (range: 1-18)

⟨aw *u*⟩

COMMAND has the effect of increasing or decreasing the inter-character white space. The program adds or subtracts specified units to the LEFT side of every text character until cancelled. (This is consistent, regardless of how the typesetter itself may handle it.) As the size of the projected character is enlarged, (as with increasing point sizes) the space between characters is proportionally enlarged and eventually becomes visually unattractive. This command allows compensation to be applied. Conversely, some type designs require "opening up" in very small sizes. First check your phototypesetter, certain models do have "Automatic Compensation".

A positive unit value after the command indicates the number of units to be added to each character. A negative value directs the number of units to subtract from each. The same amount of space is applied to all characters, regardless of their original widths. This technique is known as "White Space Reduction".

BEGINS immediately.

NOTE: The first character in a line is never altered, otherwise flush-left alignment would be inaccurate.

ENDS when a CANCEL WIDTH **⟨xw⟩** is read.

EXAMPLES of command AW

Using as a sample, the words "ALTER WIDTH" in a 12 point size, let's see the results of ranging from 6-units ADDED to 6-units SUBTRACTED from every letter. The normal line is shown with a -N.

⟨aw 5⟩ A L T E R W I D T H
⟨aw 4⟩ A L T E R W I D T H
⟨aw 3⟩ A L T E R W I D T H
⟨aw 2⟩ A L T E R W I D T H
⟨aw 1⟩ A L T E R W I D T H
⟨aw 0⟩ ALTER WIDTH-N
⟨aw-1⟩ ALTER WIDTH
⟨aw-2⟩ ALTER WIDTH
⟨aw-3⟩ ALTER WIDTH
⟨aw-4⟩ ALTER WIDTH
⟨aw-5⟩ ALTER WIDTH
⟨aw-6⟩ ALTER WIDTH

Note that only the SPACE between characters is what is affected. If a CONDENSED typeface is used, the results obtained will be proportionately reduced, since we are dealing in UNITS OF SPACE.

⟨aw 5⟩ A L T E R W I D T H
⟨aw 4⟩ A L T E R W I D T H
⟨aw 3⟩ A L T E R W I D T H
⟨aw 2⟩ A L T E R W I D T H
⟨aw 1⟩ A L T E R W I D T H
⟨aw 0⟩ ALTER WIDTH-N
⟨aw-1⟩ ALTER WIDTH
⟨aw-2⟩ ALTER WIDTH
⟨aw-3⟩ ALTER WIDTH
⟨aw-4⟩ ALTER WIDTH
⟨aw-5⟩ ALTER WIDTH
⟨aw-6⟩ ALTER WIDTH

Spacing can be mixed on the same line, and even justified. The computer keeps track of all changes. Of course, do not allow Letterspacing to occur.

This is Tight This is Normal Spacing This is Open This is Extra-Extra Tight Space This is Normal Tightened Up Normal Open-Up This is Condensed and Tight This is Normal Space

SECT. II—COMMAND STRUCTURE

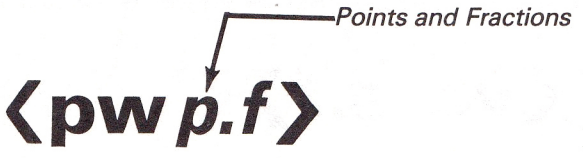
3.3 Style and Size Control Group

PW•

Point Width

POINT WIDTH

(Also called "Set Size")



COMMAND has the effect of "squeezing" or expanding a type face. The side-to-side dimension of every character in the font is changed as well as the spaces between them. This ability has been named *"rubber type"*. It behaves the same way that letters printed on a rubber band would act.

EXPANSION is created by specifying a point size *larger* than the current size.

CONDENSATION is accomplished by using a point size value *smaller* than the current size. To obtain the percentage change divide the new point width by the current size.

BEGINS immediately anywhere with a line.

ENDS

1. When another POINT WIDTH **⟨pw⟩** is issued.
2. FONT NUMBER **⟨fn⟩** also will override a previous Point Width command.

EXCEPT

1. The point width value must be a valid point size for the photo-typesetter used.
2. **This command valid only on a CRT typesetter.**

EXAMPLES of command PW

Using the same sample as on the opposite page but altering the POINT width, "expands" or "squeezes" the type itself. Note that the spaces between letters are altered in the same ratio. "Normal" is shown by —N.

⟨pw18⟩ POINT WIDTH
⟨pw17⟩ POINT WIDTH
⟨pw16⟩ POINT WIDTH
⟨pw15⟩ POINT WIDTH
⟨pw14⟩ POINT WIDTH
⟨pw13⟩ POINT WIDTH
⟨pw12⟩ POINT WIDTH—N
⟨pw11⟩ POINT WIDTH
⟨pw10⟩ POINT WIDTH
⟨pw 9⟩ POINT WIDTH
⟨pw 8⟩ POINT WIDTH
⟨pw 7⟩ POINT WIDTH
⟨pw 6⟩ POINT WIDTH

Many times only a percentage is given, let's say a 25% squeeze. That's really 75% of our 12 pt size, or 9-POINT WIDTH

⟨pw12⟩ 12-POINT WIDTH
⟨pw 9⟩ 9-POINT WIDTH....75%

That's fine. How about a 15% expansion? Starting with 12-point type, then 15% works out to 1.8 pts, so let's call it 2. Adding 12+2 makes 14 pt, like this.

⟨pw12⟩ 12-PT WIDTH
⟨pw14⟩ 14-PT WIDTH...115%

POINT WIDTHS can even be mixed on the same line

NORMAL...WIDE..NARROW

VA•

Vertical Advance

EXAMPLES of command VA

For clarity, we'll use rules instead of type for these samples. The rules are nominally spaced 10 points apart.

(1) shows several vertical advances in the same line.

(2) shows reversed leading with the same values as those in (1).

⟨vl10⟩

(1) **⟨va6⟩** **⟨va3⟩** **⟨va1⟩**

(2) **⟨va-6⟩** **⟨va-3⟩** **⟨va-1⟩**

Now consider the space between paragraphs.

Here is a small paragraph to demonstrate interparagraph spaces. There is a standard amount of white space below (two blank lines).

Here is a small paragraph to demonstrate interparagraph spaces. There is a standard amount of white space below *(Increase it 12 pts)*.

⟨va 12⟩ Here is a small paragraph to demonstrate interparagraph spaces. There is a standard amount of white space below. *(Take out 6 pts)*.

⟨va −6⟩ Here is a small paragraph to demonstrate interparagraph spaces.

VERTICAL ADVANCE Leading

(Also called "Immediate Lead", "Immediate Film Advance")

Forward - Points and Fractions

⟨va *p.f*⟩

⟨va -*p.f*⟩

Reverse - Points and Fractions

NOTE: Reverse Leading Option needed on typesetter for minus value

COMMAND adds or subtracts vertical space instantly after the command is issued, even in the middle of a word, if so desired. It is intended for usage within a line, as well as between lines.

BEGINS immediately, any place in text, where the command is sensed.

When the program encounters this command, it immediately moves the paper (or film) and does not affect the standard leading for the line. *(Compare with Vertical Extra command, 2 pages hence).*

NOTE: The VERTICAL ADVANCE command has no effect on other commands in progress which use a vertical distance parameter (such as Hangs, Skews, Length Fits, which are explained later in Section 3.7).

ENDS at the end of line in which it is issued.

EXCEPT

1. **Reverse leading (negative values) only function when phototypesetters are equipped with Reverse Leading Options.**
2. The maximum possible space is 1000 points.
3. A period must separate points from fractions.

_____ TYPESETTER 1 _____ TYPESETTER 2 CHECK if equiped with Reverse Leading Option.

SECT. III—COMPOSITION COMMANDS

3.4 Leading and Spacing Control

VB•

Vertical Back

VERTICAL BACK Leading

(Also called "Subtract Lead")

Reverse: Points and Fractions

〈vb *p.f*〉

NOTE: Reverse Leading Option needed on typesetter for this command

COMMAND is the opposite of Vertical Advance in that the film moves backwards. It is identical to a Vertical Advance with a Negative Parameter **〈va -p.f〉** command and can be used interchangeably with it. It is a matter of choice.

BEGINS immediately, anyplace.

ENDS at the end of the line in which it is issued.

EXCEPT

1. **This command works ONLY when phototypesetters are equipped with Reverse Leading Options.**
2. The maximum possible space is 1000 points.
3. A period must separate points from fractions.

TYPESETTER 1 _____ TYPESETTER 2 _____ CHECK if equiped with Reverse Leading Option.

EXAMPLES of command VB

This command directs the paper to move backwards. The example below shows that this is another way of saying VERTICAL ADVANCE NEGATIVE, the command on the previous page (ie. Lines A & B below yield the same results)

A _____ **〈va-4〉** _____

B _____ **〈vb4〉** _____

Let's try some practical examples. Commands and copy are shown first, then the results.

(1) **〈ps24〉** X **〈ps12;vb14〉** 2 **〈va14〉**

(1)..... X^{2} ↑ 14 pts up

(2) **〈ps36〉** H **〈ps14;va4〉** 2 **〈ps36;vb4〉** O

(2).. H_2O ↓ 4 pt drop

(3) **〈ps48〉** 3^{96} **〈ps10;vb 16.5〉** and up **〈va16.5〉**

(3)... 3^{96} and up ↑ 16.5 p

VE•

Vertical Extra

EXAMPLES of command VE

*This command adds (or subtracts, within limits) white space **between** lines, never within as line as is possible with the previous two commands. First, we establish a basic sample below, with built-in 12 point leading:*

ONE
TWO
THREE
FOUR
FIVE
SIX

Now we add 24 points between lines 3 and 4.

ONE
TWO
THREE

⟨ve24⟩ FOUR
FIVE
SIX

We can take out space, even if there is no reverse leading ability on the typesetter, BUT no more than 12 points. The command is limited to the current vertical advance (that's line 4). For a first try, remove 6 pts, then go the limit: 12 points

ONE
TWO
THREE
⟨ve-6⟩ FOUR
FIVE
SIX

ONE
TWO
⟨ve-12⟩ FOUR THREE
FIVE
SIX

VERTICAL EXTRA Leading

(Also called "Add Lead", "Add Film Advance")

Forward - Points and Fractions

⟨ve p.f⟩

⟨ve-p.f⟩

Reverse - Points and Fractions

NOTE: Reverse Leading Option NOT NEEDED on typesetter for minus value (if not more than current line).

COMMAND is very similiar to the Vertical Advance but effective only at the beginning of a line. It is most valuable when used between lines. It is a special command which yields additional and independent space between lines, "temporarily" in the sense that it acts for one line only.

NOTE: This command does not actually move the paper or film. The program waits for the current line leading value. Before it can act on the paper, the program alters the line leading, increasing or decreasing it by the Vertical Extra parameter. The EFFECT of reverse leading can thus be achieved without actually moving the paper backwards. The only limitation is that the program cannot subtract more than the current line leading amount.

BEGINS at the start of the current line by adding space above. Stated another way, the command is placed between lines and it changes the space between those lines.

ENDS as soon as it has been issued.

EXCEPT

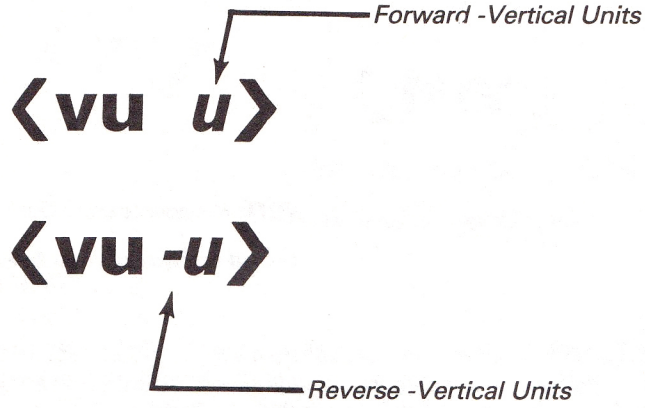
1. **Reverse leading amounts are limited to the current line leading value.**
2. Command must be used at at start of line only.
3. The maximum possible space is 1000 points.
4. A period must separate points from fractions.

VU•

Vertical Unit

VERTICAL UNITS of Leading

(Also called "Add White Space—Units")



NOTE: Reverse Leading Option needed on typesetter for minus value

COMMAND uses the same action as Vertical Advance but uses Units to express the value of the vertical spacing. There are 18 Vertical Units to an EM Square. Thus, this command will always yield a space which is proportioned to the point size. This property can be a valuable tool. Fixed relationships can thus be maintained without prior knowledge of the point sizes to be used.

BEGINS immediately.

ENDS at the end of line in which it has been issued.

EXCEPT

1. **Reverse leading (negative values) only function when phototypesetters are equipped with Reverse Leading Options.**
2. The maximum possible space is 1000 vertical units.

_____ TYPESETTER 1 _____ TYPESETTER 2 CHECK if equiped with Reverse Leading Option.

EXAMPLES of command VU

Since this command is expressed in units as the parameter, all vertical proportions can be maintained without considering point size. An excellent application is specifing the leading needed to clear descenders when changing from large to small sizes (ie. Heads-to-Body). For example, to drop down ⅓ the current point size, use 6 units (18/3).

Type

‹vu6› This copy (set 9/10) just clears the descenders of the headline because a vertical Unit command of 6 units ALWAYS provides enough extra white space regardless of headline size.

Type

‹vu6› This copy (set 9/10) just clears the descenders of the headline because a vertical Unit command of 6 units ALWAYS provides enough extra white space regardless of headline size.

Type

‹vu6› This copy (set 9/10) just clears the descenders of the headline because a vertical Unit command of 6 units ALWAYS provides enough extra white space regardless of headline size.

Type

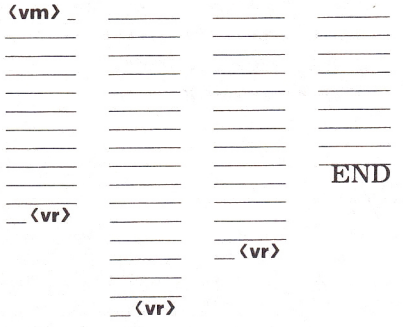
‹vu6› This copy (set 9/10) just clears the descenders of the headline because a vertical Unit command of 6 units ALWAYS provides enough extra white space regardless of headline size.

VM

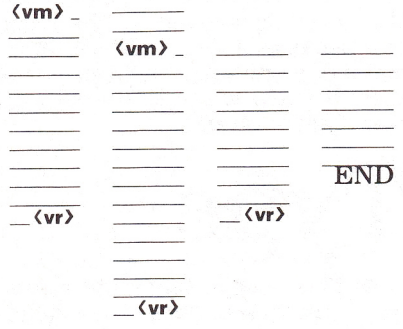
Vertical Mark

EXAMPLES of command VM

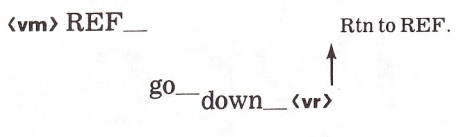
The Vertical Mark is only the first half of a story. It is a reference mark used by the Vertical Return (next page). Below four columns are simulated. They show how the copy returns each time to the Vertical Mark.

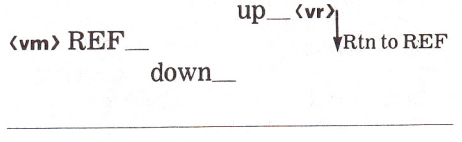


If the Vertical Mark is reissued at another spot it over rides the previous one and establishes a new point of return.



Let's try a REFERENCE LINE sample.





VERTICAL MARK

(Also called "Line Marker", "Line Pointer")

〈vm〉

NOTE: Reverse Leading Option needed on typesetter for this command

COMMAND saves the vertical position, in other words the current depth of the line in which command is given. Nothing else happens yet. When the Vertical Return command is given the paper is driven back to the Vertical Mark position. Depending upon individual condition either forward or reverse leading is used.

BEGINS at the exact vertical position issued. Once established, this point may be used as many times as desired.

ENDS only when another VERTICAL MARK **〈vm〉** is established. There is no command to cancel, since it is not necessary.

EXCEPT

1. **Reverse Leading Option to typesetting is necessary for this command to be functional.**
2. When the Vertical Mark is issued at the beginning of a line, the return will be positioned above, ready to advance paper when a line is set.
3. If command is ommitted, the top of the job will become the *"default"* value.

VR
Vertical Return

VERTICAL RETURN to Mark

(Also called "Go to Start Line")

⟨vr⟩

NOTE: Reverse Leading Option needed on typesetter for minus value

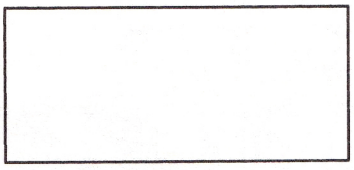
COMMAND via a forward lead or a reverse lead returns to the Vertical Mark **⟨vm⟩** that was previously issued. If no Vertical Mark was specified, the top of the job will be assumed to be it. An obvious use for a command such as this is a return up to the top of a column. Equally useful, but more subtle is to set up a datum, or reference line with **⟨vm⟩** , set a block of copy, use **⟨vr⟩** to return to base, then just repeat the process.

BEGINS at the exact point of issue.

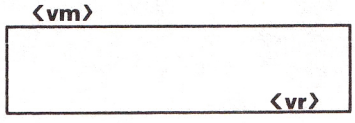
ENDS when the Vertical Mark is reached.

EXAMPLES of command VR

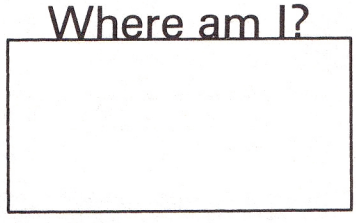
Now for a more practical application. Suppose we draw a box, using a format.



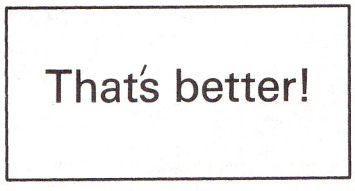
Furthermore, we can easily control the depth. To be practical we must reverse back up to the top to set type. If we build a **⟨vm⟩** *into the top line, and* **⟨vr⟩** *into the bottom line, it becomes automatic.*



Now we're ready to try with some actual type.



Forgot something! That took us back to the baseline (which is outside the box). Add some lead, and we'll fix it.

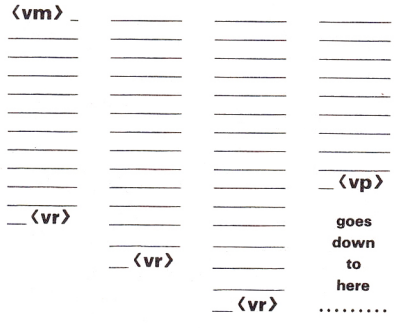


VP

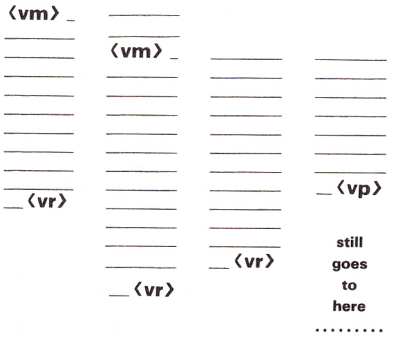
Vertical Point

EXAMPLES of command VP

Refer to the example which showed the use of the Vertical Return (see page 3-20). Command **〈vp〉** *goes to the baseline of the* **lowest** *of the* **〈vr〉** *commands issued.*



It doesn't matter if the Vertical Mark was reestablished at another spot. The Vertical point stll goes to the LOWEST point.



VERTICAL POINT

(Also called "Vertical Lowest")

〈vp〉

COMMAND via a forward leading action goes to the lowest **〈vr〉** point previously issued.

BEGINS immediately at the point of issue.

ENDS automatically after completing the leading required.

VC
Vertical Clear

VERTICAL CLEAR Leading Count
(Also called "Reset Leading Counter")

⟨vc⟩

COMMAND immediately resets the Accumulated Leading counter in program to zero. *Only* from this point on will vertical space be counted and the information shown in Compose-for-Display. This command is very useful in determining the size of a news story by itself, NOT INCLUDING the headlines, if issued at the start of the news itself. If included in the body copy formats, it becomes automatic.

BEGINS at the exact point of issue.

ENDS when the Vertical Mark is reached.

EXCEPT it is ignored:

1. If the galley mode option is in effect at the time it is issued. *(See Galley Define* ⟨**gd**⟩ *command.)*

EXAMPLES of command VC

(1) Shown below is a newspaper headline followed by a story. The headlines occupy lines 002, 003 and 004. At 005 there is a blank line. The story begins at 006 and the depth at this point is 034 or 3.4 inches.

```
1P323PHA    EDITOR ACTIVE
LN DEP DEF  LINES:00011  INCHES:00004.1  PICAS:0
001 000 10  ↵
001 000 28  ⟨H3⟩■↵
002 013 01  Headline for the story↵
003 021 03  below goes here and↵
004 029 09  takes three lines↵
005 033 28  ■↵
006 034 01  ↵Copy for story starts here. When
007 038 01  a VERTICAL CLEAR is placed on
008 037 01  the line above the accumulated
009 038 01  depth column (DEP) second at
010 040 01  left clears and begins counting
011 041 02  again from that point on.↵
```

(2) If a ⟨**vc**⟩ *command is placed on line 006, observe that the depth figure has been reset to zero. The depth counter now starts anew. The first line of the story now reads 001.*

```
1P323PHB    EDITOR ACTIVE
LN DEP DEF  LINES:00011  INCHES:00000.8  PICAS:0000
001 000 10  ⟨fdH3;cw30;nr2,36,48⟩↵⟨cw10,19,10;FE⟩↵
001 000 10  ↵
001 005 28  ⟨H3⟩■↵
002 013 01  Headline for the story↵
003 021 03  below goes here and↵
004 029 09  takes three lines↵
005 033 28  ■↵
006 001 01  ↵⟨vc⟩Copy for story starts here. When
007 002 01  a VERTICAL CLEAR is placed on
008 004 01  the line above the accumulated
009 005 01  depth column (DEP) second at
010 005 01  left clears and begins counting
011 008 02  again from that point on.↵
```

HP•

EXAMPLES of command HP

This command can control very fine movements. In the lines below the space between boxes is being increased by ONE point on each line. The command is actually placed between boxes, but is shown at the front of each line so you may see the exact space produced. Remember, the period indicates points.

⟨hp.0⟩ □□ Space normal.
⟨hp.1⟩ □ □ Normal + 1 pt.
⟨hp.2⟩ □ □ Normal + 2 pt.
⟨hp.3⟩ □ □ Normal + 3 pt.
⟨hp.4⟩ □ □ Normal + 4 pt.
⟨hp.5⟩ □ □ Normal + 5 pt.
⟨hp.6⟩ □ □ Normal + 6 pt.
⟨hp.7⟩ □ □ Normal + 7 pt.
⟨hp.8⟩ □ □ Normal + 8 pt.
⟨hp 1⟩ □ □ Plus 1 Pica.

Space can also be subtracted by using a negative number. The same commands as above with MINUS values look like this.

⟨hp.-0⟩ □□ Space normal.
⟨hp.-1⟩ □□ Normal less 1 pt.
⟨hp.-2⟩ □□ Normal less 2 pt.
⟨hp.-3⟩ □□ Normal less 3 pt.
⟨hp.-4⟩ □□ Normal less 4 pt.
⟨hp.-5⟩ □□ Normal less 5 pt.
⟨hp.-6⟩ □□ Normal less 6 pt.
⟨hp.-7⟩ □□ Normal less 7 pt.
⟨hp.-8⟩ □□ Normal less 8 pt.
⟨hp -1⟩ □□ Normal less 1 Pica.

HORIZONTAL Space in PICAS

(Also called "Controllable Fixed Space")

SPACE: Picas and/or Points

⟨hp *p.p*⟩

⟨hp *-p.p*⟩

BACKSPACE: Picas and/or Points

COMMAND instructs the program to insert a specified fixed space (forward or backward) at the current point in the textstream. It is expressed in absolute units of measurement, namely Picas and/or Points, providing a method of securing the SAME FIXED SPACE regardless of the point size used.

BEGINS AND ENDS immediately at the point issued, anywhere within the line.

EXCEPT

1. The maximum positive space is 1000 points.
2. The maximum negative space may not exceed beyond the left margin of the line where issued.

SECT. III—COMPOSITION COMMANDS

3.4 Leading and Spacing Control

HU•

Horizontal Units

HORIZONTAL Space in UNITS

(Also called "Kerning Units" "Unit Spaces")

Space: Relative Units

⟨hu *u*⟩

⟨hu -*u*⟩

Backspace; Relative Units

COMMAND is similiar to the previous page except that relative units of measurement are used, namely 1/18 of an EM space. Here then is an opportunity to specify a non-justifying space which is PROPORTIONAL to type size, other than the common EM, EN or THIN spaces. Backspacing equal to or based on character width, with this command, becomes a simple procedure.

BEGINS AND ENDS immediately at the point issued, anywhere within the line.

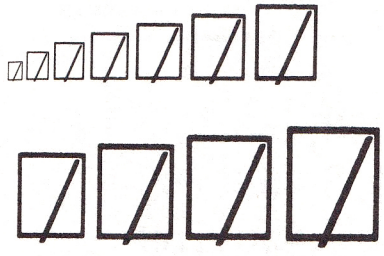
EXCEPT:

1. The maximum positive space is 1000 units.
2. The maximum negative space may not exceed beyond the left margin of the line where issued.

EXAMPLES of command HU

The advantage of this command is that it is independent of point size. Let's say we want to create a special character, by exposing one symbol on top of the other. Here a □ and / are combined by "backspacing". The SAME horizontal unit command is used in ALL cases, just the point size is changed.

COPY
□ **⟨hu-12⟩** /
RESULT:



This command can also be used for "kerning", in this manner:

COPY
(0) WALTER (normal)
(1) W **⟨hu-1⟩** AL **⟨hu-1⟩** TER
(2) W **⟨hu-2⟩** AL **⟨hu-2⟩** TER
(3) W **⟨hu-3⟩** AL **⟨hu-3⟩** TER

0 WALTER
1 WALTER
2 WALTER
3 WALTER

EM/EN/TH

SINGLE KEY COMMAND SYMBOL

EXAMPLES of command EM/EN/TH

A common use of fixed spaces is to allow numerals to align.

□□□.....Em spaces

□□□.....En spaces

□□□.....Thin spaces

Take the following line of numbers and punctuation and let's replace certain characters with fixed spaces (shown by boxes). The last line shows the actual space in place.

123,456:789-10.
1□3,456:789-□.
1 3,456:789- .

123,456:789-10.
123□4□□:789-10.
123 4 :789-10.
123,456:789-10.

□□□456:789-10.
456:789-10.

EM/EN/THIN Fixed Spaces

(Also called "MUTT", "Em Quad", "NUT", "En Quad")

⟨em⟩ or □ **Single-Key Called "EM SPACE"**

⟨en⟩ or □ **Single-Key Called "EN SPACE"**

⟨th⟩ or □ **Single-Key Called "THIN SPACE"**

COMMAND or SINGLE KEY inserts a Fixed Space (EM, EN, or THIN) into the textstream immediately. The prime value of these spaces is that they are in relative units and are proportional to the point size (ie., when the point size changes, so do they, and in exactly the same ratio).

EM SPACE is a fixed space normally equal to the square of the point size and is *always* 18 units. It is often used as an indent for paragraphs. (An EM in 9 point type is 9 pts high x 9 pts wide and *and at the same time* 18 units wide x 18 units high.)

NOTE: In condensed or expanded styles, the EM varies accordingly so that, for example, a condensed Gothic EM is only 16 units wide, while 18 units high (ie. a "condensed square").

EN SPACE is always equal to the width of a numeral (and often referred to as a "figure space"). It is used to indent and align figures when they are in columns.

NOTE: It is usually half an EM (9 units) in newspaper typefaces, but not necessarily so: (Helvetica numerals, and thus also the EN, are 10 units wide, not the normal 9).

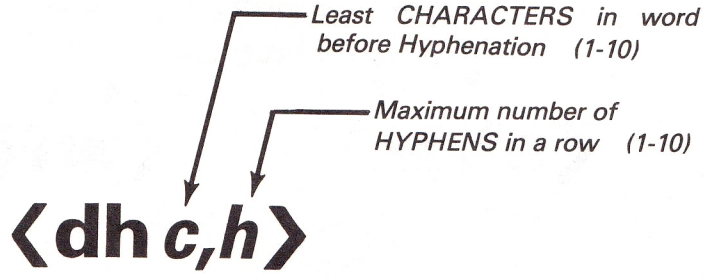
THIN SPACE is always equal to the width of a period or comma and is used in aligning figures when they have decimal points or commas in large numbers.

NOTE: It is usually one third of an EM (6 units) but not always so. (Helvetica periods and commas are only 5 units wide, not the normal 6).

DH
Define Hyphenation

DEFINE HYPHENATION Mode

(Also called "Hyphenation Set-Up")



COMMAND sets up the conditions for hyphenation to occur. The shortest word which allows hyphenation routine to take place is specified by the value of "*c* " which may vary from a 1- to 10-character word. Too many hyphenated lines in a row mar the appearance of a column of type. The program checks the value of "*h* " (range 1-10) and inhibits any further hyphens-in-a-row, above the number specified.

BEGINS immediately and lasts for entire job.

ENDS if another DEFINE HYPHENATION **⟨dh⟩** is given or new job started.

EXCEPT

1. If no **⟨dh⟩** is specified or new story is started the *"default"* values stored in "HSETUP" are used. Minimum word size is commonly five characters, and 2 hyphens-in-row is used often.

EXAMPLES of command DH

The same story is presented here twice. Note the word "after" at the end. Each sample shows the effect if:

(1) Only 6-CHAR WORDS or greater will be hyphenated. 4-HYPHENS-IN-A-ROW are the maximum allowed.

(2) Only 4-CHAR WORDS or greater will be hyphenated. 2-HYPHENS-IN-A-ROW are the maximum allowed.

⟨dh6,4⟩
(1) This paragraph was designed to demonstrate to you the hyphenation and the justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, and after & after and after & after and after & after.

⟨dh4,2⟩
(2) This paragraph was designed to demonstrate to you the hyphenation and the justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, after & after and after & after and after & after.

AH

Allow Hyphenation

CANCEL WITH
XH

EXAMPLES of command AH, XH

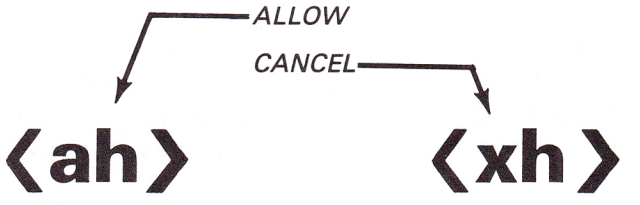
The same story is presented twice, thus;
(1) CANCEL HYPHENATION command before copy. Note the very large spaces between words
(2) ALLOW HYPHENATION command before copy. Note the improved appearance and that it takes two lines less

〈xh〉
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins
CANCEL.....14 lines.

〈ah〉
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum space-bands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins
ALLOW.....12 lines.

ALLOW HYPHENATION Mode

(Also called "Hyphenation On")



COMMAND to "allow" causes all words which overset the column width to be submitted to the hyphenation program. A breakpoint resulting in the tightest line is determined by using the following priorities:

1. Natural or Discretionary Hyphens
2. Exception Dictionary Breakpoints
3. Rules of logic

ENDS when a CANCEL HYPHENATION **〈xh〉** is read.

EXCEPT

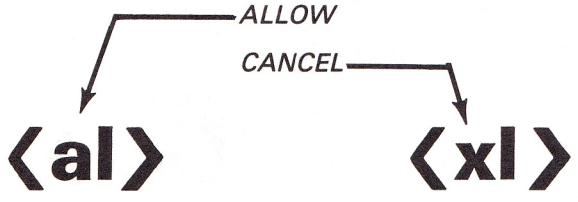
1. The program assumes hyphenation unless otherwise specified in the "HSETUP" file.
2. The **〈ah〉** command is normally not used except after a **〈xh〉** command to re-establish hyphenation.
3. If a cancel **〈xh〉** is in effect, compound words will still be broken at hyphens.
4. Any lines containing only one word will force a breakpoint, even if it is incorrect, unless cancel **〈xh〉** is in effect.

CANCEL WITH
XL

AL

ALLOW LETTERSPACE Mode

(Also called "Letterspace On")



COMMAND to "allow" permits words to be letterspaced if the line cannot be justified when spacebands are expanded to the maximum allowable value. If "cancel" is issued all line expansion must take place within the spacebands regardless of maximum value.

BEGINS immediately.

ENDS when a CANCEL LETTERSPACE ⟨xl⟩ command is read.

EXCEPT

1. The program assumes letterspacing unless otherwise specified in the "HSETUP" file.
2. The ⟨al⟩ command is normally not used except after a ⟨xl⟩ command to re-establish letterspacing.

EXAMPLES of command AL, XL

The same story is presented twice, to dramitize; (1) CANCEL LETTERSPACE command before copy. Note the very large spaces between words (2) ALLOW LETTERSPACE command before copy. Note the word space has now been distributed within words.

⟨xl⟩
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins

⟨al⟩
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum space-bands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins

AD

Allow Dictionary

CANCEL WITH
XD

EXAMPLES of command AD, XD

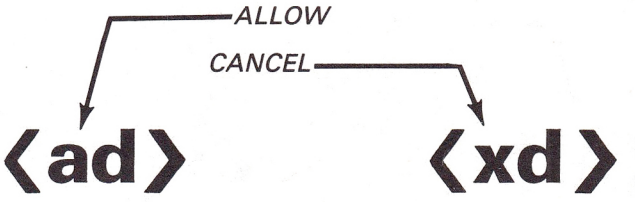
Normally there is no need to turn off the exception dictionary. However, a question sometimes arises as to whether or not a particular word is contained in the dictionary listing. Of course, one could search the file "HDCTION", but this can be tedious with nearly 3,000 words to look through.
A faster way is to
(1) CANCEL DICTIONARY and then repeat the word several times (to exercise all break points by logic rules), then,
(2) ALLOW DICTIONARY and observe if incorrect breaks are fixed.

〈xd〉
(1) biography biography biography biography biography
.....INCORRECT
.....bio-graphy
...Not in dictionary

〈ad〉
(2) biography biography biography biography biography
.....CORRECT
.....bi-ogra-phy
.....In dictionary

ALLOW DICTIONARY Mode

(Also called "Use Exception Dictionary")



COMMAND to "allow" permits the hyphenation routine to first consult the "EXCEPTION" Dictionary. Words which are susceptible to hyphenation error are entered in this dictionary with proper breakpoints. In this way the program can be "taught" not to repeat any hyphenation mistakes. Approximately 3000 words can be accomodated.

BEGINS immediately.

ENDS when a CANCEL DICTIONARY **〈xd〉** is read.

EXCEPT

1. The program assumes dictionary is wanted, unless otherwise specified in the "HSETUP" file.
2. The **〈ad〉** command is normally not used except after a **〈xd〉** command to re-establish Exception Dictionary look-up.

HO
HOmograph

HOMOGRAPH

(Also called "Discretionary Hyphen", "Allowable Break Point")

⟨ho⟩

COMMAND specifies the positions of allowable breakpoints within a word (where the program may hyphenate if the need arises). Homographs (commonly known as "Discretionary Hyphens"), are used when unusual or highly technical words crop up.

Sometimes compound words will not break at the most natural places. For instance, *underwater* will split only at *un/derwa/ter*, but not at *under/water*. Use the command **⟨ho⟩** to overcome such problems.

When only one Homograph is used in a word, it becomes a mandatory breakpoint (assuming it does not fall in the overset region of a line). Because of this property, it has great value as method of correcting erroneous hyphenation points, by forcing a word to split at a designated spot. A natural hyphen is dangerous to use for this purpose. Some prior change may have moved the word in question to the middle of the line. The natural hyphen will now show up when not wanted, but a homograph is invisible.

When one Homograph is used, and it is used *before* the word, then hyphenation is inhibited and the entire word is carried to the next line, ignoring both the dictionary and logic break points. This is handy for proper nouns, such as geographical or personal names.

BEGINS only if needed. Use as many **⟨ho⟩** 's in a word as desired. Only one can be converted to a hyphen at the end of a line.

EXAMPLES of command HO

If you know that a word is not in the exception dictionary, then use Homographs to establish the proper break points.

COPY: abscound Carter

Abscound abscound abscound Carter Carter Carter Carter

COPY: abs ⟨HO⟩ cound Car ⟨HO⟩ ter

Abscound abscound abscound Carter Carter Carter Carter

If you want to FORCE hyphenation at a particular spot in a word, then a single Homograph becomes a mandatory break point. Thus, un/derwa/ter vs. under⟨HO⟩ water

underwater underwater underwater underwater
vs
underwater underwater underwater underwater underwater underwater

To inhibit hyphenation in a single word, such as a Name or Proper Noun, place the Homograph before the word. Thus, Do/na/van vs ⟨HO⟩ Donavan:

Donavan Donavan Donavan Donavan Donavan Donavan
vs
Donavan Donavan Donavan Donavan Donavan Donavan

AN•

Alter miNimum band

CANCEL WITH
XN

EXAMPLES of command AN, XN

The same story is presented thrice, thus;
(1) ALTER MINIMUM to 12 units. Note the wide spaces between words, actually excessive, but little variation from line-to-line.
(2) ALTER MINIMUM to 9 units. That's a little better. Quaded line shows exact minimum spacing.
(3) CANCEL MINIMUM . The 'default' value of 4 units is automatically restored. Note now, the line-to-line variations.

(1) **‹an12›**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum space-bands exert a tremendous influence.
This is a quad left line

(2) **‹an8›**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence .
This is a quad left line

(3) **‹xn›**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence .
This is a quad left line

ALTER MINIMUM Spaceband Width

(Also called "Change MIN Band")

Units (Range: 0-100)

‹an *u*›

COMMAND redefines the minimum amount of space between words measured in Relative Units. In the calculations for justification, this value is used first as the trial wordspace. It is then allowed to expand to achieve line justification. Quadded lines, however, use the MIN value as a fixed space between words. It is also used in this manner as the last line of a paragraph. If there is too great a variation between "tight" and "loose" lines increase the MIN width value so that it is closer to the MAX width.

BEGINS at the start of a line. This command may be issued only at the beginning of a line.

ENDS when a **‹xn›** is read. The "Font Table" value is substituted. *(Fill in values for your system in the table on next page.)*

EXCEPT

1. The value MIN must be between 0 and 100 units.
2. The MIN value must always be less than the MAX.
3. If used in middle of line, command will be ignored.
4. If a change in the font number or in the point size is made, with either of these commands the program refers to the "Font Tables" (HFONT) and re-establishes the min/max band values stored there. To be effective the **‹an›** and **‹am›** commands should therefore follow either a size or font change.
   (See next page for values stored in Font Tables.)

CANCEL WITH
XM

AM•

SECT. III—COMPOSITION COMMANDS

3.5 Hyphenation and Justification Control

Alter Maximum band

ALTER MAXIMUM Spaceband Width

(Also called "Change MAX Band")

Units (Range: 4-100)

⟨am *u*⟩

COMMAND redefines the maximum amount of space between words measured in Relative Units. If letterspacing is allowed (see command **⟨al⟩** for information) the maximum spaceband width will not be exceeded by the program. Additional space will be distributed between letters in the line. If letterspacing is not allowed (see command **⟨xl⟩** for information) then the spaceband will continue to expand beyond the maximum specified until the line is justified.

BEGINS at the start of a line. This command may be issued only at the beginning of a line.

ENDS when a **⟨xm⟩** is read. The "Font Table Value" value is substituted.

EXCEPT

1. The value MIN must be between 0 and 100 units.
2. The MAX value must always be more than the MIN.
3. If used in middle of line command, will be ignored.
4. If a change in the font number or in point size is made, with either of these commands, the program refers to the "Font Tables" (HFONT) and re-establishes the min/max band values stored there. To be effective the **⟨an⟩** and **⟨am⟩** commands should therefore follow either a size or font change. *(See values below.)*

MINIMUM BAND	MAXIMUM BAND	
_____	_____	FONT 1 or GRID 1
_____	_____	FONT 2 or GRID 2
_____	_____	FONT 3 or GRID 3
_____	_____	FONT 4 or GRID 4
_____	_____	FONT 5 or GRID 5
_____	_____	FONT 6 or GRID 6
_____	_____	FONT 7 or GRID 7
_____	_____	FONT 8 or GRID 8

EXAMPLES of command AM, XM

The same story is presented twice;
(1) ALTER MAXIMUM to 10 units. Note the amount of letterspacing that takes place.
(2) CANCEL MAXIMUM . The 'default' value of 20 units is automatically restored. Letterspacing is no longer needed to justify.

(1) **⟨am10⟩**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins
This is a quad left line

(2) **⟨xm⟩**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins
This is a quad left line

RL/RR

Ragged Left/Right

CANCEL WITH
RX

EXAMPLES of command RL, RR

The same story is presented thrice, thus;
(1) NORMAL: Uses 'default' commands of AH, AL from 'HSETUP'
(2) RAGGED LEFT: Note that ALL spaces between words is at the minimum value. All hyphenation has been suppressed.
(3) RAGGED RIGHT: Note this sample is identical with the ragged left above it except the lines are uneven at the right instead.

(1) **(Normal)**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins.
This is a quad left line

(2) **⟨rl⟩**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins.
This is a quad left line

(3) **⟨rr⟩**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins.
This is a quad right line

RAGGED LEFT/RAGGED RIGHT

(Also called "Flush Right-Uneven Left"/"Flush Left-Uneven Right")

⟨rl⟩ **⟨rr⟩**

COMMAND causes textstream to be set with either the left or right margins uneven or ragged. The copy is set with the minimum spaceband used between words. No letterspacing is allowed. The hyphenation program is automatically turned OFF when any of the three ragged commands are used. If a less ragged effect is desired then allow hyphenation to take place by using the command **⟨ah⟩** to override the automatic shut-off.

BEGINS immediately.

ENDS when a RAGGED CANCEL **⟨rx⟩** command is issued. This command in turn, then automatically restores the hyphenation program.

EXCEPT when using the RAGGED CENTER, paragraphs must be ended with a Quad Center, rather than a End Paragraph.

CANCEL WITH **RX**

RC

SECT. III—COMPOSITION COMMANDS

3.5 Hyphenation and Justification Control

Ragged Center

RAGGED CENTER

(Also called "Uneven Left and Right")

‹rc›

COMMAND is the same as the previous page except that BOTH left and right margins are made ragged. It is not a true random uneveness but rather centered lines of varying lengths.

All the ragged commands operate to break lines as though hyphenless justification decisions are being made, and then the expansion of spacebands is inhibited. The excess space is moved left, right or divided equally as the case may be.

BEGINS immediately.

ENDS when a RAGGED CANCEL **‹rx›** command is issued. This command in turn, then automatically restores the hyphenation program.

EXCEPT when using the RAGGED CENTER, paragraphs must be ended with a Quad Center, rather than a End Paragraph.

EXAMPLES of command RC

(4) RAGGED CENTER: Note that this is line-for-line the same as examples (2) & (3), but centered symetrically, that is, the degree of "raggedness" of the right and left margins is the same.

(5) RAGGED CENTER with Hyphenation: Note that now there is a lesser degree of irregularity. That is why hyphenation is automatically cancelled with this command.

(4) **‹rc›**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins
This is a quad left line

(5) **‹rc;ah›**
This paragraph was designed to demonstrate to you the hyphenation and justification control commands. You will be able to see the effects of allowing hyphenation, letterspacing and exception dictionary. The minimum and maximum spacebands exert a tremendous influence on appearance. Finally, copy will be seen with ragged margins
This is a quad right line

SINGLE KEY COMMAND SYMBOL

JU

JUstify

EXAMPLES of command JU

The justify command spreads the line to fill the the line length in effect. The **〈ju〉** *is applied at the end of each line below. Note, that as copy is removed the word spaces increase to spread the line. Letterspacing has been cancelled.*

1 Now is the time to see justify
2 Now is the time to see just
3 Now is the time to see
4 Now is the time to
5 Now is the time
6 Now is the

This time we will allow letterspacing to occur. Compare with the results above. The maximum spaceband is set at 8. Note: Letterspacing starts at line 5

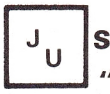
1 Now is the time to see justify
2 Now is the time to see justif
3 Now is the time to see justi
4 Now is the time to see just
5 Now is the time to see jus
6 Now is the time to see ju
7 Now is the time to see j
7 Now is the time to see

Compare with the results above. The maximum spaceband is set at 16 Letterspacing now at line 7

1 Now is the time to see justify
2 Now is the time to see justif
3 Now is the time to see justi
4 Now is the time to see just
5 Now is the time to see jus
6 Now is the time to see ju
7 Now is the time to see j
8 Now is the time to see

JUSTIFY Line

(Also called "Return" or "Elevate")

〈ju〉 or 

Single-Key Called "JUSTIFY"

COMMANDS or SINGLE KEYS in this group determine the manner in which a line will behave. Normally the HCS program automatically determines the proper line break.

The output typesetter carries out justification so that the line is filled out from the left to right margins. Situations do arise which require a particular spot to be a line end. Use the **〈ju〉** command for this. It is translated on output to the line ending code for your typesetter.

BEGINS & ENDS immediately. It ends the current line (or tab field) at the point issued. All preceding text on the line or tab column is then spaced to full measure.

SINGLE KEY COMMAND SYMBOL

QL/QC/QR

QUAD LEFT/QUAD CENTER/QUAD RIGHT

(Also called "Flush Left/Center/Right", "Copy Left/Center/Right")

⟨ql⟩ or ← **Single-Key Called "COPY LEFT" "QUAD LEFT"**

⟨qc⟩ or ↔ **Single-Key Called "COPY CENTER" "QUAD CENTER"**

⟨qr⟩ or → **Single-Key Called "COPY RIGHT" "QUAD RIGHT"**

COMMAND causes whatever text has thus far been assembled on a line to be flushed to the left, to the right or centered in the column. The spaceband will remain at the minimum value. Letterspacing is prohibited by the program.

NOTE: You cannot "skip-a-line" by using just a Quad Code. *(See EXCEPT below.)*

This command can be used in either a column of text (set up by a **⟨cw⟩** command) or a tab column (these set-ups are explained in Section 3.8)

NOTE: The HCS program does NOT use special "TAB QUADS". In fact, ALL commands in this section for a Text column work in a Tab column.

BEGINS immediately as soon as the command is read by the program. The program then notes the amount of space remaining in the line and then:

1. Leaves it at the end of the line for a QUAD LEFT, **⟨ql⟩** or
2. Moves it to the start of line for a QUAD RIGHT, **⟨qr⟩** or
3. Divides the space equally and places it before and after the copy for a QUAD CENTER, **⟨qc⟩** .

EXCEPT that any line ending signals **⟨ql⟩** , **⟨qc⟩** , **⟨qr⟩** or **⟨ju⟩** by themselves on a line with **no** text will result in **no film advance**. This has been done purposely. In Compose-for-Display a blank line **does** occur. If a blank line is also wanted on the typesetter, then use fixed spaces, (EN's) or (EM's). They do not show but the program treats them as text characters, resulting in a line of space.

EXAMPLES of command QL, QC, QR

The sample below shows Quads in action in a standard text column.

Copy goes left. ⟨ql⟩
Here's a quad left line. ⟨ql⟩
Flush on left. ⟨ql⟩

Copy center. ⟨qc⟩
Now a centered line. ⟨qc⟩
Ragged left/right. ⟨qc⟩

Copy goes right. ⟨qr⟩
Here's a quad right line. ⟨qr⟩
Flush on right. ⟨qr⟩

LEFT
CENTER
RIGHT

Headline Flush Left on 3 lines

Headline Centered on 3 lines

Here are Quads used in tabular work to both quad and advance to next col.

QL	QL	QL	QL	QL	
QL	QL	QL	QL	QL	
QL	QL	QL	QL	QL	
QC	QC	QC	QC	QC	
QC	QC	QC	QC	QC	
QC	QC	QC	QC	QC	
QR	QR	QR	QR	QR	
QR	QR	QR	QR	QR	
QR	QR	QR	QR	QR	

QP

Quad Paragraph

SINGLE KEY COMMAND SYMBOL

¶

QUAD PARAGRAPH

(Also called "End Paragraph")

〈qp〉 or ¶ **Single-Key Called "END PARA"**

COMMAND ends current line with a Quad Left and automatically inserts an indent at the start of the next line. The amount of indent is determined by the values shown later under command Allow Automatic Paragraph Indent **〈aa〉** . *(See page 3-49.)* The single keystroke command "End Para" is used very often, and therefore the key on the terminal has been made oversize.

BEGINS immediately.

EXCEPT

1. This command will not indent if the CANCEL AUTO Paragraph Indent **〈xa〉** has been issued.
2. If the command is given with no text, it will cause only the left margin to be reset and then an indent on the next line.

EXAMPLES of command QP

The "END PARA" key is probably the most used command on the keyboard. It's popular because it's automatic and it's easy. Indents for the "next paragraph" are under HCS program control (See 'AA', page 3-49). Let's see this command in-between paragraphs of varying point sizes.

.....and now we come to the ending and the very last line of a paragraph. 〈qp〉
Starting a new paragraph we find that the indent is one EM.....

.....and now we come to the ending and the very last line of a paragraph. 〈qp〉

Starting a new paragraph we find that the indent is one EM...

All the indents above were one EM. Below, as the column width increases, so does the amount of paragraph indent, although the point size remains fixed at 9 point.

1 EM .....from the couple's small house in Norristown recently. In the beginning. 〈qp〉

Just short of three weeks into the ordeal Jim broke. "If I had to do this for the rest of my life I'd go bananas." .....

2 EM's.....he was indeed superhuman. "Julia Child, Dr. Spock, and Rudolph Valentino all rolled into one." reflected Jim's pretty wife Louise from the couple's small house in Norristown recently. In the beginning. 〈qp〉

Just short of three weeks into the ordeal Jim broke. "If I had to do this for the rest of my life I'd go bananas." came the surrender in writing, a notarized declaration of defeat. The document continues: "Granted, women today .....

3 EM's.....two months ago 30-year old Jim Shingler accepted the awesome challenge of house hold husband. And in the beginning, it is true, he was indeed superhuman. "Julia Child, Dr. Spock, and Rudolph Valentino all rolled into one." reflected Jim's pretty wife Louise from the couple's small house in Norristown recently. In the beginning. 〈qp〉

Just short of three weeks into the ordeal Jim broke. "If I had to do this for the rest of my life I'd go bananas." me e surrender in writing, a notarized declaration of defeat. The document continues: "Granted, women day have tomatic washers, perma press clothing, and.....

SINGLE KEY COMMAND SYMBOL

QM
Quad Middle

QUAD MIDDLE
(Also called "Flush Left & Right", "Insert Space")

〈qm〉 or **Single-Key Called "W/SPACE"**

COMMAND causes any text *preceeding* to set flush left and any text *following* to set flush right. Because the action is like a gigantic spaceband, this operation is often called "Insert Space Here". There is no limit to the number of **〈qm〉** commands within a line. The total available space is divided equally among them.

BEGINS & ENDS when line is completed. Any line ending signal may be used; **〈ql〉** , **〈qc〉** , **〈qr〉** , **〈ju〉** or **〈qp〉** . All have the same effect. More than one QUAD MIDDLE command can used on a line. Each **〈qm〉** used fills equally with space to justify the line.

EXCEPT

1. Only **one** (1) QUAD MIDDLE **〈qm〉** should be used in a line, if there is an Indent Hang **〈ih〉** command starting the line.

EXAMPLES of command QM

The single-key QUAD MIDDLE is shown in the COPY below by a box ☐ The last three examples have multiple commands per line. Note that space is distributed evenly among them.

COPY
1234 ☐ ABCDEF
123456789 ☐ ABC
12 ☐ AB
12345678 ☐ ABCDEFGHI
1234 ☐ 5678 ☐ ABCDEFGHI
12345 ☐ 78 ☐ ABCD ☐ EFGHI
12345 ☐ 78 ☐ ABCD ☐ EF ☐ GHI

EXAMPLE
1234 ABCDEF
123456789 ABC
12 AB
12345678 ABCDEFGHI
1234 5678 ABCDEFGHI
12345 78 ABCD EF GHI

Spacebands are always held at their minimum values. The examples here demonstrate this by varying the minimum band value from 2 to 8 units.

COPY
Now is the ☐ time for all good

EXAMPLES
Now is the time for all good
Now is the time for all good
Now is the time for all good
Now is the time for all good
Now is the time for all good
Now is the time for all good
Now is the time for all good

SINGLE KEY COMMAND SYMBOL

WL

With Leaders

EXAMPLES of command WL

This command works like the Quad Middle, except dots (actually periods) are inserted, instead of spaces. Here we show what happens if some of the copy is left out of the line. A □ indicates the command.

COPY
Copy Left □ Copy Right
Copy Left □
□ Copy Right
□
One □ Two □ Three □ Four

EXAMPLES
Copy Left.....Copy Right
Copy Left .....
.....Copy Right
.....
One.....Two.....Three.....Four

Next, we demonstrate that ANY line ending signal will work.

COPY
Quad L □ Ending **⟨ql⟩**
Quad C □ Ending **⟨qc⟩**
Quad R □ Ending **⟨qr⟩**
Justify □ Ending **⟨ju⟩**
Quad P □ Ending **⟨qp⟩**

EXAMPLES
Quad L.....Ending
Quad C.....Ending
Quad R.....Ending
Justify .....Ending
Quad P.....Ending

Finally, let's mix commands on the same line.

COPY
QM **⟨qm⟩** WL **⟨wl⟩** WR **⟨wr⟩**
WL **⟨wl⟩** WR **⟨wr⟩** QM **⟨qm⟩**
WR **⟨wr⟩** WL **⟨wl⟩** WR **⟨wr⟩** ETC.

EXAMPLES
QM WL.....WR_____
QM WL.....WR_____
WL.....WR_____QM
WR_____WL.....WR_____
WL.....WR_____WL.....
QM WL.....QM

Fill Line WITH LEADERS

(Also called "Dotted Rule")

⟨wl⟩ or

Single-Key Called "W/..."

COMMAND fills the line with a dotted leader wherever issued. (The HCS program uses periods.) Copy before the leader is flushed to the left, copy after is flushed right. A spaceband is used just before the first dot, to justify the line exactly. (Space is always less than one dot, otherwise additional dots are used instead.) There is no limit to the number of **⟨wl⟩** commands in a line. Any other Insert or Fill command may be mixed with this command.

BEGINS & ENDS when line is completed. Any line ending signal may be used; **⟨ql⟩** , **⟨qc⟩** , **⟨qr⟩** , **⟨ju⟩** or **⟨qp⟩** . All have the same effect. More than one Fill WITH LEADER command can used on a line. Each **⟨wl⟩** used fills equally with dots to justify the line.

EXCEPT

1. Only **one** (1) Fill WITH LEADERS **⟨wl⟩** should be used in a line, if there is an Indent Hang **⟨ih⟩** command starting the line.

SECT. III—COMPOSITION COMMANDS

3.6 Line Control Group

WR
With Rules

Fill Line WITH RULES

(Also called "Draw Rule")

⟨wr⟩

COMMAND works the same as "With Leaders" except that the BLR (Base Line Rule) is used as the fill character. Used by itself on a line, it draws a rule from margin to margin. The thickness of the line is normally dependent on the point size, but sometimes the type style has an effect. Use your own typesetter to determine the exact thickness for each point size and/or font.

NOTE: Most CRT typesetters draw rules by moving a beam spot across the tube rather than by flashing a line segement character. Thickness is determined by point size, which is usually accomplished by "wiggling" the beam spot a controlled amount. *(See samples at right).*

BEGINS & ENDS when line is completed. Any line ending signal may be used; **⟨ql⟩** , **⟨qc⟩** , **⟨qr⟩** , **⟨ju⟩** or **⟨qp⟩** . All have the same effect. More than one Fill WITH RULES command can used on a line. Each **⟨wr⟩** used fills equally with rules to justify the line.

EXCEPT

1. Only **one** (1) Fill WITH RULES **⟨wr⟩** should be used in a line, if there is an Indent Hang **⟨ih⟩** command starting the line.

EXAMPLES of command WR

This command will draw a line from margin-to-margin when used alone on a line. The rules below were produced on a CRT typesetter, with first a point size, then a rule command, then a line end.

⟨ps12;wr⟩ 1/2pt rule
⟨ps18;wr⟩
⟨ps24;wr⟩ 1 pt rule
⟨ps30;wr⟩
⟨ps36;wr⟩ 1 1/2 pt rule
⟨ps48;wr⟩ 2 pt rule
⟨ps54;wr⟩
⟨ps60;wr⟩
⟨ps66;wr⟩ 3 pt rule
⟨ps72;wr⟩

Now a simple form using rules. Note that if several rules are used on the same line, they are all of equal length. The "signed" line was done with an indent.

Name_____
Address_____
City_____ State_____ Zip_____
Phone_____ Apt.No._____
Quant._____ Style_____ Size_____ Color_____

*Signed*_____

WX•

With "X" Character

EXAMPLES of command WX

First let's make special rules using symbols, like this.

Star ☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆
Bullet •••••••••••••••••••••••••••••••••
Box □□□□□□□□□□□□□□□□□□□□
Cap W WWWWWWWWWWWWWWWWW
l.c m mmmmmmmmmmmmmmmmmmmm

Now by using the Alter Width command, see how the appearance can be controlled.

Literally thousands of leaders can be created by cross-breeding symbol, point size and alter widths. Here are a few using only periods as the 'X' character.

⟨ps10;pw4;wx⟩ •
10 pt size, 4 pt width

⟨ps6;pw8;wx⟩ •
6 pt size, 8 pt width

⟨ps14;pw9;wx⟩ •
14 pt size, 9 pt width

⟨ps14;pw8;aw-2;wx⟩ •
14 pt size, 8 pt width, alter width -2

⟨ps14;pw22;wx⟩ •
14 pt size, 22 pt width

⟨ps14;pw22;aw-2;wx⟩ •
14 pt size, 22 pt width, alter width -2

⟨ps8;pw2;wx⟩ •
8 pt size, 2 pt width

⟨ps8;pw5;wx⟩ •
8 pt size, 5 pt width

⟨ps18;pw24;aw-1;wx⟩ •
18 pt size, 24 pt width, alter width -1

Fill Line WITH "X" CHARACTER

(Also called "Define Leader Character")

Any Text Character in the Font

⟨wx⟩ *X*

COMMAND is just like the prior "Fill" commands, but now we have our choice of any character available. Also, by using the **⟨aw⟩** Alter Width we can change the spacing for special effects.

The character can be any printable one, be it lower case, upper case or Pi symbol.

BEGINS & ENDS when line is completed. Any line ending signal may be used; **⟨ql⟩** , **⟨qc⟩** , **⟨qr⟩** , **⟨ju⟩** or **⟨qp⟩** . All have the same effect. More than one Fill WITH 'X' CHARACTER command can used on a line. Each **⟨wx⟩** used fills equally with x' characters to justify the line.

EXCEPT

1. Only **one** (1) Fill WITH 'X' CHARACTER **⟨wx⟩ *X*** should be used in a line, if there is an Indent Hang **⟨ih⟩** command starting the line.

CANCEL WITH
IX

IT•

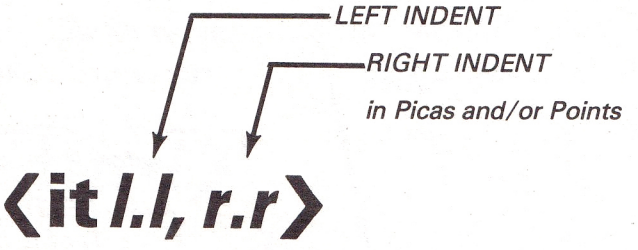
SECT. III—COMPOSITION COMMANDS

3.7 Indent Control Group

Indent Take

INDENT TAKE

(Also Called "Indent Left or Right", "Column Indents")



COMMAND is used to set the margins of the column. Either the left, right, or both margins can be indented with just this one command. If either the left or right parameter value is omitted, the program assumes it is zero, i.e. that margin is not affected. Therefore, this single command serves as Indent Left, Indent Right or Indent Both.

BEGINS at the start of *next line.*

ENDS

1. With a CANCEL INDENT **⟨ix⟩** command.
2. With a CANCEL ALL Tabs & Indents **⟨xx⟩** command.
3. With another INDENT TAKE **⟨it⟩** command.
4. At the end of a job.

EXCEPT

1. Comma must separate parameters if both are present.
2. Comma may be omitted if right parameter is absent.
3. Period must separate picas from points in parameter.

EXAMPLES of command IT

This command affects either the left, right or both margins. The samples show:
(1) Normal col. of 11 picas, no indents
(2) Indents: 3 left, 2 right
(3) Indents: 2 right only
(4) Indents: 3 left only

(1) **(Normal)**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits.

(2) **⟨it 3,2⟩**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(3) **⟨it ,2⟩**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits.

(4) **⟨it 3⟩**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits.

HI•

CANCEL WITH
HX

Hang Indent

EXAMPLES of command HI

The samples below give the command first and then the results.

(1) Normal col of 11 picas,

(2) Hang: Picas-Left 3,-Right 1, Lines-Wait 3, -Depth 4

(3) Hang: Three set-ups. Notice use of zeros at first, then that the zeros are dropped, but NOT commas.

(4)Try to follow this one yourself.

(1) **Normal**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

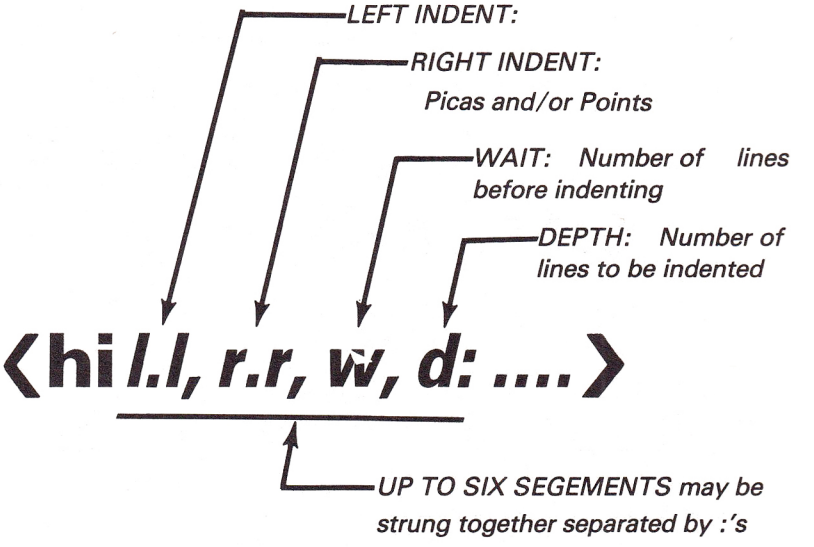
(2) **‹hi3,1,3,4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(3) **‹hi1,0,1,1 : 2,0,0,2 : 3.6,,,3 : 4.2,,,4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control command.

(4) **‹hi1,1,1,1 : 2,2,,3 : ,3,1,2 : 3,1,1›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, auto-

HANG INDENT

(Also called "Indent After 'X' Lines")



COMMAND sets up conditions for hanging indents as shown above. A maximum of six (6) Hang command parameter segments may be strung together, separated by colons (:) and terminated with an End Command Code (›). Each segment of the command is satisfied in turn, before the next one goes into effect.

BEGINS immediately if parameter value "*w*" is zero. Otherwise, indents start after the number of lines in "*w*".

ENDS

1. With a CANCEL HANG **‹hx›** .
2. With CANCEL ALL Tabs/Indents **‹xx›** .
3. Another HANG INDENT **‹hi›** will override.
4. If parameter "*d*" is zero (0), the Hang Indent value continues indefinitely. Otherwise the indent terminates after the number of lines in "*d*".

EXCEPT

1. Commas must separate individual parameters.
2. Periods must separate picas and points.
3. Omitted parameter values will be assumed zero (0). Do NOT omit any commas.
4. Period must separate picas from points in parameters.

SECT. III—COMPOSITION COMMANDS

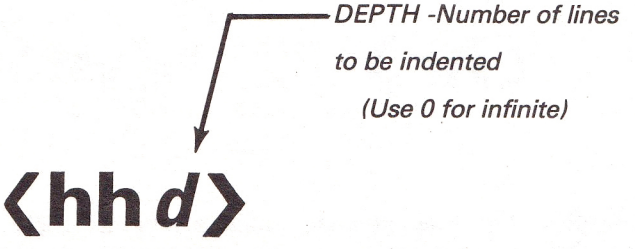
CANCEL WITH
HX

HH
Hang Here

3.7 Indent Control Group

HANG HERE

(Also called "Indent Text" "Hang Text")



COMMAND sets up a Left Hang Indent based on the width of text preceeding it.
The line containing the command will be set to current column width. Subsequent lines will be indented at the point where the Hang Here command was inserted. Any place within a line is permissible.

NOTE: This command is NOT cumulative. If a second Hang Here is issued the program will compute, by using the previous copy WITHOUT the previous Hang Here value. *(See examples at right.)*

BEGINS on the *next line.*

ENDS

1. With a CANCEL HANG **⟨hx⟩** .
2. With CANCEL ALL Tabs/Indents **⟨xx⟩** .
3. Another HANG HERE **⟨hh⟩** will override.
4. If parameter "*d* " is 0, the Hang Indent value continues indefinitely. Otherwise the indent terminates after the number of lines in "*d* ".

EXAMPLES of command HH

Wherever the Hang Here is issued becomes the indent value. The command **⟨hh⟩** *when used below is shown by an* **X**

Here now is a **X**paragraph designed to demonstrate the indent control commands.

If a second Hang Here is inserted, it will NOT include the previous Hang. Note the second **X** *produces a hang indented the amount of the word* You *only.*

Here now is a **X**paragraph designed to demonstrate the indent control commands. You **X** will be able to see the effects of hang indents, and of length fits.

An unusual feature is that the hang can be stopped after a specified number of lines. The **X** *now represents* **⟨hh3⟩** *thus:*

Here now is a **X**paragraph designed to demonstrate the indent control commands. You will be able to see the effects of hangs.

(1)...The count ability makes it easy to produce 3-line Initial Caps using an **⟨hh3⟩** *command. First without reverse lead.*

A N INITIAL Capital letter can be automatically accomodated by using reverse leading and the Hang Here command as shown here.

(2)...Now add the proper amount od reverse leading.

A N INITIAL Capital letter can be automatically accomodated by using reverse leading and the Hang Here command as shown here.

(3)...Finally check, with different copy.

WITH INITIAL Capital letter that can be automatically accomodated by using reverse leading and the Hang Here command.

HR

EXAMPLES of command HR

The Hang Indent and Hang Here are "one shot" commands. Once used up, they must be rewritten and reissued. This command avoids all of that. Below is shown
(1st) The Hang Command + Text
(2nd) Text only
(3rd) Hang Repeat + Text + imbedded Hang Repeat.
Note the second Hang Repeat causes the command to start again before completion of the first.

(1) **⟨hi 3,1,2,4⟩ + Text**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(2) **Text Only**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(3) **⟨hr⟩ + Text + ⟨hr⟩ again**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the ⟨hr⟩ effects of hang indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control commands.

HANG REPEAT Indent

(Also called "Recycle Hang")

⟨hr⟩

COMMAND will temporarily terminate either a Hang Indent or Hang Here at the current line and then cause that command to repeat. In other words, each time the Repeat Hang is read a full width line is output and then the Hang Indent is restarted.

BEGINS at the start of NEXT LINE.

ENDS according to Hang Indent instructions in effect or if there are several **⟨hr⟩** 's, then the cycle repeats and repeats.

CANCEL WITH **IR**

IH

SECT. III—COMPOSITION COMMANDS

3.7 Indent Control Group

Indent Hang

INDENT HANG Copy

(Also Called "Generate Indent")

⟨ih⟩

COMMAND sets up a special column composed of the "white space" created by a Quad Middle (or Fill With) command. Put another way:

- The Left Indent is equal to the Flush Left Copy
- The Right Indent is equal to the Flush Right Copy
- The Space left in the Middle becomes the column in question.
- The command itself must be issued at the *START* of the Quad Middle line.

NOTE: To make this command more practical, an additional space of four points has been intentionally added to both the left and right indents.

BEGINS on the *next line* following the command.

ENDS

1. With a INDENT HANG RESET **⟨ir⟩** command.
2. With a CANCEL ALL Tabs & Indents **⟨xx⟩** command.
3. At the end of a job.

CAUTION:

Unlike most other set-up type commands issuing another INDENT HANG *will not override* the previous widths set up. You *must* first reset with a **⟨ir⟩** command.

EXCEPT the command will be ignored:

1. If there is more than one Quad Middle (or Fill) in the line. The program obviously cannot make a multiple choice of white space.

EXAMPLES of command IH

This is a very special kind of command. The examples graphically show how it works. Note that there is an extra 4 pts of space automatically built- in on both sides.

COPY:
⟨ih⟩ LEFT ⟨wl⟩ & RIGHT ⟨ju⟩

RESULT:

LEFT.....& RIGHT
This copy is now being set in the space generated by a Fill with Leaders command above.

COPY:
⟨ih⟩ 1,2 ⟨wr⟩ 3,4 ⟨ju⟩

RESULT:

1,2_____3,4
This copy is now being set in the space generated by a Fill with Rule command above.

Something fancier? This time the original line uses a Quad Middle to establish the column.

COPY:
⟨ih;cw13;va22⟩ THE ⟨qm⟩ CO. ⟨ju⟩
⟨cr2,36,9;pw24⟩ HENDRIX ⟨ju⟩
⟨cr2,8,8⟩ The word HENDRIX above was set in
...etc.

RESULT:

THE HENDRIX CO.
The word HENDRIX above was set in 36 pt type on 9 leading, then justified. This centered it vertically, then this copy followed on the same measure.

A most practical application is for grocery ads:

STEAK CHOICE N.Y. CUT **2.29**

TEA LIPTON'S FLOW-THRU **89¢**

IP•

Indent Paragraph

CANCEL WITH
QP or ¶

EXAMPLES of command IP

This command sets up for just one paragraph, if the paragraph mark is used to terminate it. Below is shown
(1) Standard paragraph.
(2) INDENT PARAGRAPH command issued. Note that there is no first line auto indent. Copy ended with a Quad Left.
(3) Indent still in effect. Copy now ends with Quad Paragraph.
(4) Normal paragraph reappears.

(1)This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

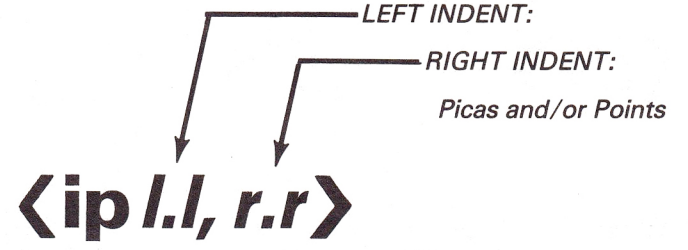
‹ip1.4,2.9›
(2)This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits. **‹ql›**

(3)This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits. **‹qp›**

(4)This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

INDENT PARAGRAPH

(Also called "Paragraph Space")



COMMAND sets up left and/or right indents for a paragraph that follows it, assuming that the ends of paragraphs are defined by the single-key "End Para" or with the **‹qp›** command.

However, if Quad Lefts are used to end the paragraph, this command remains in effect and continues to the next paragraph.

BEGINS on the NEXT LINE after the command is given.

ENDS

1. When when a QUAD PARA **‹qp›** or single-key "End Para" is read.
2. It is also terminated by a CANCEL ALL Tabs/Indents **‹xx›** .

CAUTION: Indent values cannot exceed column widths.

1. Commas must separate individual parameters.
2. Periods must separate picas and points.
3. Omitted parameter values are assumed zero (0). Do NOT omit any commas.
4. Period must separate picas from points in parameters.

SECT. III—COMPOSITION COMMANDS

CANCEL WITH
XA

AA
Allow Automatic

3.7 Indent Control Group

ALLOW AUTOMATIC Paragraph Indent

(Also called "Standard Paragraph Indent")

⟨aa⟩

COMMAND enables the Automatic Paragraph Indent table values to become operative, when the single key "End Paragraph" or 2-letter **⟨qp⟩** commands are issued. The amount of indent (in EN's) is determined by the line length. Because it is expressed in EN's, note that it is proportional to type size . A typical set-up such as the one shown below can be part of the HCS program.

FROM 0 to 10.6 picas: INDENT - 2 ens
FROM 11 to 19.6 picas: INDENT - 3 ens
FROM 20 to 35.6 picas: INDENT - 4 ens
FROM 36 to 45.0 picas: INDENT - 5 ens

BEGINS immediately. If not issued the program assumes **⟨aa⟩** is on.

ENDS with a CANCEL AUTO INDENT **⟨xa⟩** , in which case **no** indents follow the single-key "End Paragraph" or **⟨qp⟩** command.

TABLE: Enter values for your own Automatic Paragraph Indent Table below:

FROM _____ to _____ picas: INDENT - ____ ens
FROM _____ to _____ picas: INDENT - ____ ens
FROM _____ to _____ picas: INDENT - ____ ens
FROM _____ to _____ picas: INDENT - ____ ens

EXAMPLES of command AA

For several examples of this command in action, look at the **⟨QP⟩** *command on page 3-38.*

Below is shown the effect of cancelling and then restoring this command.

(1) CANCEL Auto Indent.
(2) ALLOW Auto Indent.

Note the indent changed the amount of copy that fit on the first line.

⟨xa⟩
The paragraph you are reading demonstrates how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the typesetter.

⟨aa⟩
The paragraph you are reading demonstrates how the Hendrix 6400 System accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the typesetter.

LL•

Length Left

CANCEL WITH
LX

EXAMPLES of command LL

This is similiar to Hang (but works only for the left margin). The BIG difference depth is in absolute (picas), not relative (lines). You get the SAME size notch with different point sizes.
(1) Normal col of 12 picas,
(2) Notch—4⁹ wide x 3⁴ deep, 9 pt size
(3) Notch—Same, but 6 pt size
(4) Notch—Same, but 12 pt size
(5) Six Length Left commands grouped into one

(1) **Normal**
This paragraph was designed to demonstrate to you the indent control commands. You will see the effects of length fits.

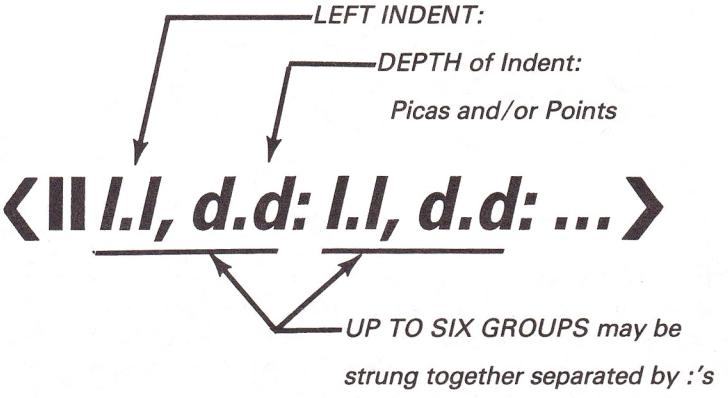
(2) **‹LL4.9,3.4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, indents, and length fits.

(3) **‹LL4.9,3.4;ps6›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits, which is being demonstrated right now.

(4) **‹LL4.9,3.4;ps12›**
This paragraph was designed to demonstrate to you the indent control commands.

(5) **‹LL1,1:2,2:3,3:2,2:1,1›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control command.

LENGTH Fit LEFT Indent



COMMAND sets up a left indent for a specified vertical depth. The left indent remains effective until
(1) The vertical depth expires,
(2) A new Length Left is given or
(3) A cancel command of some kind is employed.

Up to six Length Left indents may be combined into a single command. Each is separated from the others by colons. Each portion is satisfied in turn before the next one starts. This allows the side of a type column to be fitted around odd shapes. Length Left indents can be added to other indents that are active (Columns, Skews, Hangs etc).

BEGINS immediately if parameter value "*d*" is zero. Otherwise, indents start after the depth in "*d*".

ENDS

1. With a LENGTH LEFT CANCEL **‹lx›** .
2. With CANCEL ALL Tabs/Indents **‹xx›** .
3. Another LENGTH LEFT **‹ll›** will override.
4. If parameter "*d*" is 0, the Length Left value continues indefinitely. Otherwise the indent terminates after the Pica and/or Point depth in "*d*".

EXCEPT

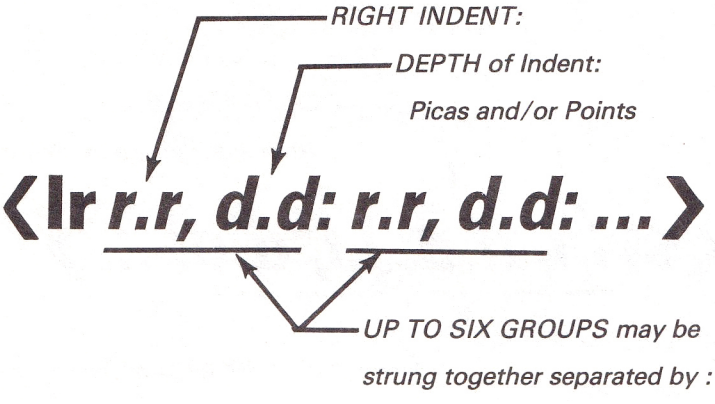
1. Commas must separate individual parameters.
2. Periods must separate picas and points.
3. Colons must separate groups.

CANCEL WITH
RX

LR•
Length Right

SECT. III—COMPOSITION COMMANDS
3.7 Indent Control Group

LENGTH Fit RIGHT Indent



COMMAND sets up a right indent for a specified vertical depth. The right indent remains effective until

(a) The vertical depth expires,
(b) A new Length Right is given or
(c) A cancel command of some kind is employed.

Up to six Length Right indents may be combined into a single command. Each is separated from the others by colons. Each portion is satisfied in turn before the next one starts. This allows the side of a type column to be fitted around odd shapes. Length Right indents can be added to other indents that are active (Columns, Skews, Hangs etc).

BEGINS immediately.

ENDS

1. With a LENGTH RIGHT CANCEL **‹rx›** .
2. With CANCEL ALL Tabs/Indents **‹xx›** .
3. Another LENGTH RIGHT **‹lr›** will override.
4. If parameter "*d*" is 0, the Length Right value continues indefinitely. Otherwise the indent terminates after the Pica and/or Point depth in "*d*".

EXCEPT

1. Commas must separate individual parameters.
2. Periods must separate picas and points.
3. Colons must separate groups.

EXAMPLES of command LR

This command is identical to the previous command but acts on the RIGHT margin. Both can be used together.
(1) Normal
(2) Notch—Same, but on right 9 pt size
(3) Notch—Same, but 6 pt size
(4) Six Length Left AND Right commands grouped into one

(1) **Normal**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of length fits.

(2) **‹LR4.9,3.4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(3) **‹LR4.9,3.4;ps6›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits, which is being demonstrated right now.

(4) **‹LL2,2:3,3:4,4:3,3:2,2:1,1; LR2,2:3,3:4,4:3,3:2,2:1,1;›**
This paragraph was designed to demonstrate to you the indent control commands. You now will be able to see the effects of hanging indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control command.

SL•

Skew Left

CANCEL WITH
SX

EXAMPLES of command SL

Skews are still another form of indent control. Here we show:
(1) Skew IN—Use 4 picas
(2) Skew OUT—Use minus 4 picas
(Note first line is full measure)
(3) Skew IN & OUT
(Note last line IN is duplicated as the first line OUT)
(4) Length Fit AND Skew added

(1) **‹SL4,4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

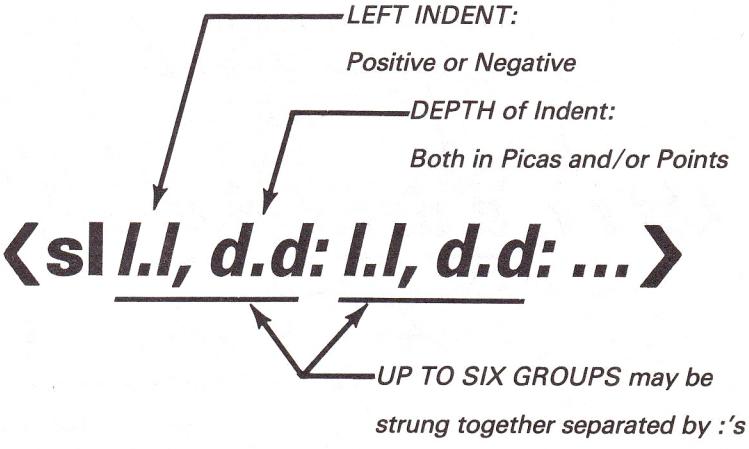
(2) **‹SL-4,6›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(3) **‹SL4,4:-4,4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

(4) **‹LL3,7;SL4,4›**
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits.

SKEW LEFT Indent

(Also called "Slant Indent")



COMMAND sets a slanted left margin. The first line is set to a full measure. Each succeeding line is indented a greater amount. When the depth specified is reached, the left indent asked for will be reached.

The direction of the slant can be reversed by using a minus indent value (use hyphen for minus sign). Now, the first line has the full amount of indent, and each line thereafter is indented LESS, until full column width is achieved at the depth in the command.

NOTE: The paragraph above is not entirely accurate in one respect, intentionally. With a negative value for the indent (reverse skew direction) the FIRST line is equal to the PREVIOUS line length. It is of importance only when a Skew Out follows a Skew In or some other indent.

BEGINS to indent at the start of the NEXT LINE and with the previous line length. Up to six skew values may be strung together, separated by colons (:).

ENDS

1. When the last of all combined skews is reached.
2. With a SKEW CANCEL **‹sx›** .
3. With CANCEL ALL Tabs/Indents **‹xx›** .
4. Another SKEW LEFT **‹sl›** will override.

EXCEPT

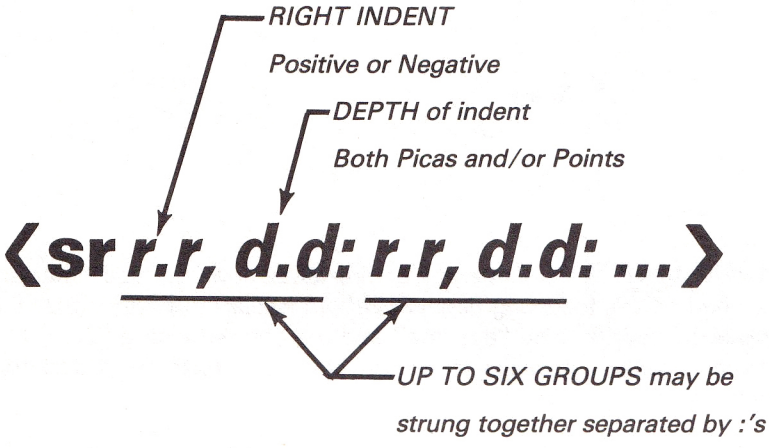
1. Commas must separate individual parameters.
2. Periods must separate picas and points.
3. Colons must separate groups.

CANCEL WITH
SX

SR•
Skew Right

SECT. III—COMPOSITION COMMANDS
3.7 Indent Control Group

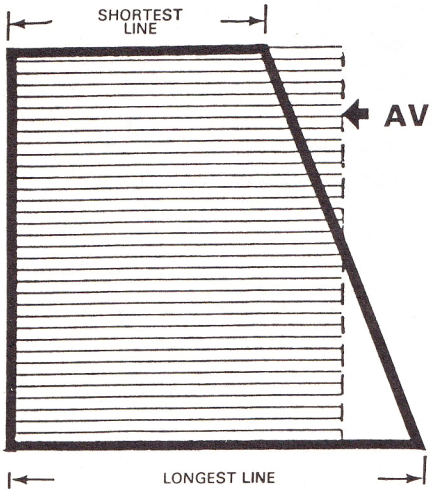
SKEW RIGHT Indent



COMMAND acts the same as previous page except it applies to the right margin. Skews may be combined with other indent commands to obtain compound effects.

COPYFIT for skewing by first finding the average length line, thus.

$$\frac{\text{Shortest line PLUS longest line}}{2}$$



1. Set block of copy using average line length.
2. Copy fit to desired length.
3. Apply skew commands for final effect.

EXAMPLES of command SR

Skews again, but on the right. Here we have run out of copy before the skew parameters were completed.

⟨SR3,9⟩
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and fits. . . . . .
. . . . . .
. . . . . .

Several skews in a row, all with the same depth, but increasing indents. Note the accurate slope control.

⟨SR 1,4 : 3,4 : 5,4⟩
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control commands. This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews.

Now we try a "waiting game". Both Length and Skew Right have a 'zero' indent value for 3 pica depth. See Italic nos. In this way you can "start" the command downstream.

⟨LR *0,3*: 2,8; SR*0,3*: 2,4: -2,4⟩
This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits. This paragraph was designed to demonstrate to you the indent control commands. You will be able to see the effects of hang indents, skews, automatic paragraph indents, and length fits. This paraaph was designed to demonstrate to

TA

Tab Activate

CANCEL WITH
TX

EXAMPLES of command TA, TX

These commands are used as pairs in order to turn the tab mode on and off.
In the samples below each time ITALIC type appears the tab mode has been turned off by a command **〈tx〉** *. When tabular appears again, it has been restarted by a command* **〈ta〉**

HIGH TIDE TODAY

Plum Island	____	12:18
BOSTON	____	12:16
Plymouth	0:00	12:23

Here are some lines of type that are not in tabulation mode as can be seen by the margins.

Cape Cod Canal	8:42	9:01
Falmouth	____	12:00

Maybe an explanation of some sort would be used at this point in the tabular material.

Chatham	0:25	12:48
New Bedford	8:52	9:14

Here is a paragraph of newspaper copy that comes after having turned off the tab mode. The heading below is still in galley mode.

HIGH TIDE TOMORROW

Plum Island	____	12:18
BOSTON	____	12:16
Plymouth	0:00	12:23
Cape Cod Canal	8:42	9:01
Falmouth	____	12:00

Here's another interruption in the tabular material.

Chatham	0:25	12:48
New Bedford	8:52	9:14
Cape Cod Canal	8:42	9:01
Falmouth	____	12:00
Chatham	0:25	12:48
New Bedford	8:52	9:14

TAB ACTIVATE Mode

(Also called "Tab Mode")

〈ta〉

COMMAND activates any of the tabulation "set-ups" that are described on the following pages. *Without issuing this command, tabbing* **does not work**. The "activate" and the corresponding "cancel" commands make it possible to switch back and forth from Tab to Text Modes.

The Tab Set-ups that are enabled with a TAB ACTIVATE **〈ta〉** command are:

1. Tab Stops.
2. Tab Widths.
3. Tab Equal.
4. Tab Proportional.
5. Tab Indent.
6. Tab Text.

NOTE: The tabulation set-up is NOT destroyed when a cancel is issued. It will be re-activated when the **〈ta〉** is re-issued.

BEGINS at the beginning of the NEXT LINE.

ENDS

1. With a TAB CANCEL **〈tx〉** command.
2. With a CANCEL ALL Tabs/Indents **〈xx〉** command.
3. With the start of new job.

TK
Tab Kill

TAB KILL Remaining Columns

⟨tk⟩

COMMAND causes the program to stop and "kill" any remaining columns on the line. It skips to the NEXT line and starts at the first column.

NOTE: It is not necessary to issue a line ending code AFTER this command.

CAUTION: If you wish to skip an individual or several tab columns, it requires a Quad (or line-ending code) for each column passed over. Command **⟨tk⟩** can only be used to end a line.

BEGINS immediately.

EXCEPT if tabbing is not activated the command will be ignored, and no error message output.

EXAMPLES of command TK

Many times the last few columns of a table are used infrequently. Tabbing requires that a line ending signal be supplied for EACH column. With this command its easy to start a new line from ANY column.

COMPANY RATINGS

	1977	1976	1975	1974	1973	1972	1971	1970
Acme	24	12	89	13	4	26	73	92
Brown	15	37	8	48	51	7		
Davis	7	73	26					
Edison	118	52	39	69	25	6	60	31
Electra	156	2	38	29	4			
Frost	58	36						
Grant	72	61	85	33	54	27	84	41
Jarvis	5	37	8	48				

TS•

Tab Stops

CANCEL WITH
TX

ON THE NEXT PAGES dealing with tabbing, the same examples will be used over and over, but each time using a different tab set-up technique. This way you can concentrate on the type of tab command rather than copy. For instance, all example 1's will use the same text (ie., 1S, 1W, 1E, etc.) The first line in each table will identify the columns by number and show their boundaries. The last line again shows the boundaries.

EXAMPLES of command TS

This command sets up tab "STOPS", much like a typewriter, except the ***right*** *margin is set. Note that the first value in the command below, namely* **2**, *is an indent. The last value is a* **12**, *which means the last column ends at 12 picas. This is beyond our 11 pica type column margin, but allowable with this command.*

Here is some copy to judge the size of a standard 11 pica column of type before we set up tabs.

(1S) **‹ts2,4,6,8,10,12›** + **‹ta›**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

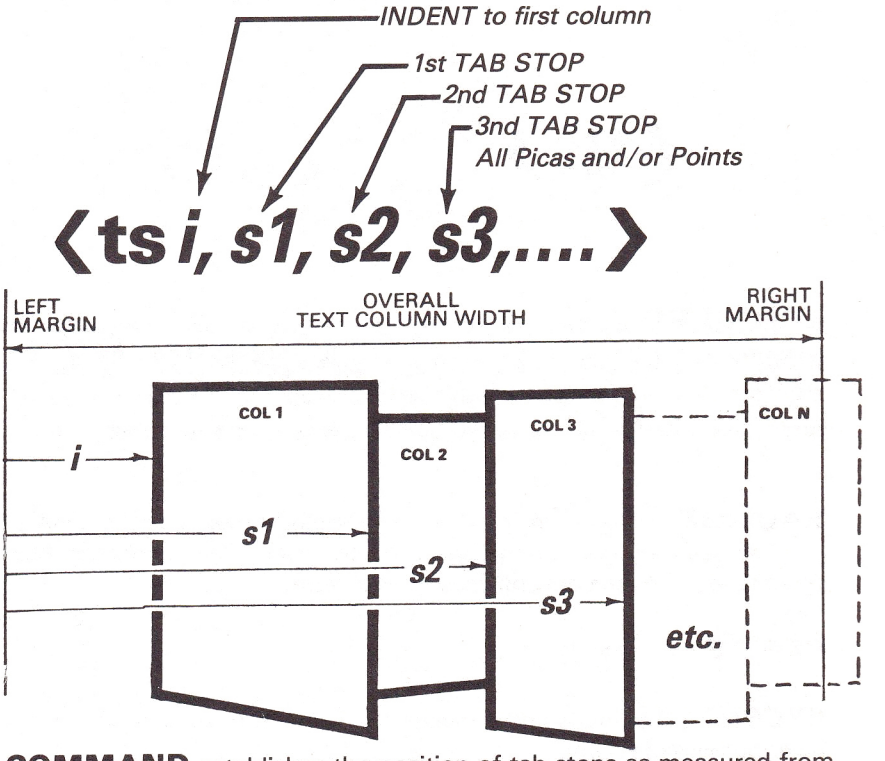
Now, what if the indent is left out? Since the stops can't move, the extra space is absorbed into the first column. Watch for this! It's different in the other commands.

(2S) **‹ts,4,6,8,10,12›** + **‹ta›**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

TAB STOPS

(Also Called "Tabbing Points")



COMMAND establishes the position of tab stops as measured from the left margin, in much the same way as a typewriter sets stop points on its tab bar mechanically. Be sure to note that the first parameter (*i*) is an indent which goes to the start of the first tab column. From there on, each stop value sets up the **right margin** of a column. A stop may fall anyplace. It may be greater or it may be less than the current text column width. Any line control commands, which operate in a text column, are valid in a tab column. *(See Section 3.6 Line Control Group.)*

BEGINS when the TAB ACTIVATE **‹ta›** is given. Any line-ending signal causes an advance to the NEXT COL. The last signal (equal to the number of columns) automatically starts a new line.

ENDS

1. With a TAB CANCEL **‹tx›** command.
2. With a CANCEL ALL Tabs/Indents **‹xx›** command.
3. When another TAB STOP **‹ts›** command overrides.
4. When a TAB WIDTH, EQUAL, PROPORTIONAL or TEXT set-up is issued.

EXCEPT

1. Commas must separate column parameters.
2. Periods must separate picas and points.
3. 20 COLUMNS is the maximum.

CANCEL WITH
TX

TW•

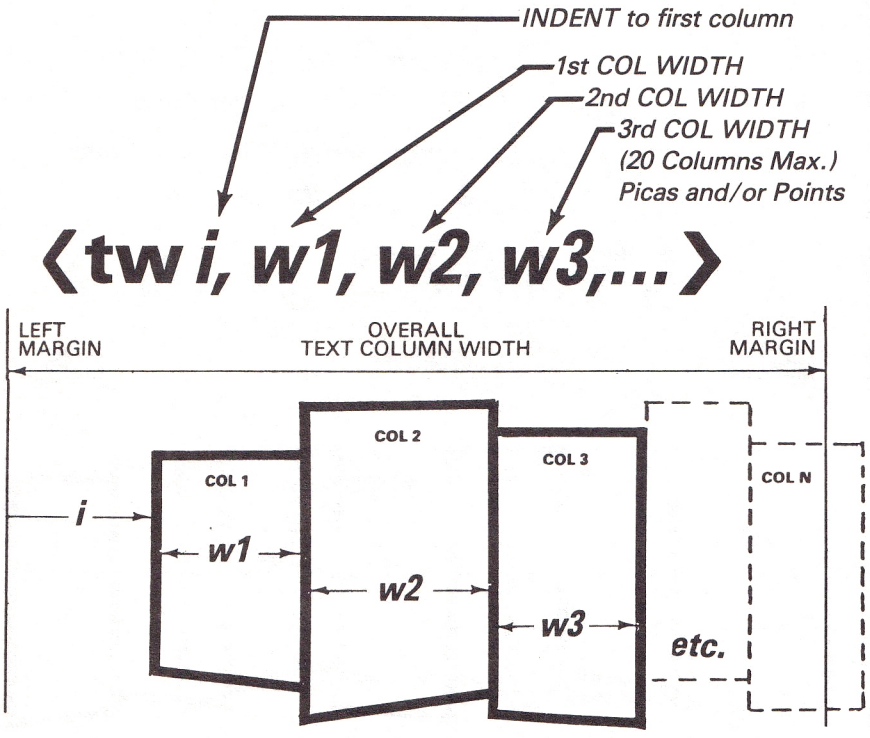
SECT. III—COMPOSITION COMMANDS

3.8 Tabbing Control Group

Tab Widths

TAB WIDTHS

(Also called "Set in Columns")



COMMAND establishes tabular columns by specifying the width of each column in picas and/or points. Note that the first value is an *indent* to the start of the first column width. If zero is used for the indent value, first column will start *at the left margin. (Compare with Tab Stops).*

BEGINS when the TAB ACTIVATE **⟨ta⟩** is given. Any line-ending signal causes an advance to the NEXT COL. The last signal (equal to the number of columns) automatically starts a new line.

ENDS

1. With a TAB CANCEL **⟨tx⟩** command.
2. With a CANCEL ALL Tabs/Indents **⟨xx⟩** command.
3. When another TAB WIDTH **⟨tw⟩** command overrides.
4. When a TAB SET, TAB EQUAL, TAB PROPORTIONAL or TAB TEXT set-up is issued.

EXCEPT

1. Commas must separate column parameters.
2. Periods must separate picas and points.
3. 20 COLUMNS is the maximum.

EXAMPLES of command TW

This command sets up cols using "WIDTHS". Let's duplicate the the first example (1S) on the opposite page. The indent and each column size was 2 picas wide. Thus, in this command all parameters are 2!

Here is some copy to judge the size of a standard 11 pica column of type before we set up tabs.

(1W) **⟨tw2,2,2,2,2,2⟩** + **⟨ta⟩**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

Now if the indent is left out, it's gone! The col widths still remain constant, but they start at the LEFT MARGIN and the entire table shifts. Watch out for this!

(2W) **⟨tw,2,2,2,2,2⟩** + **⟨ta⟩**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

If we want to exactly duplicate the second sample (2S) on the opposite side, we must ADD the indent to the first col . Like this.

(3W) **⟨tw,4,2,2,2,2⟩** + **⟨ta⟩**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

TE•

Tab Equal

CANCEL WITH
TX

EXAMPLES of command TE

This command sets up EQUAL columns. Again we duplicate, examples (1S) & (1W). All we need to do is ask for 5 columns. This time however, the text column width WILL determine the final tab col. So that the results will be consistent, change the copy paragraph below to 12 picas wide.

Here is some copy to judge the size of a *Special 12 PICA* column of type before we set up tabs.

(1E) **‹te2,5›** + **‹ta›**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

Left out the indent again? It seems to be gone. Not really. A new wrinkle! Its been distributed into ALL columns equally

(2E) **‹te,5›** + **‹ta›**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

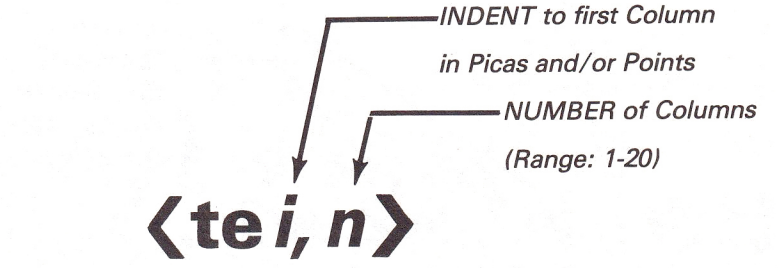
Now let's try to duplicate sample (2S) again. We can't! This command handles EQUAL columns only and the first col is larger.

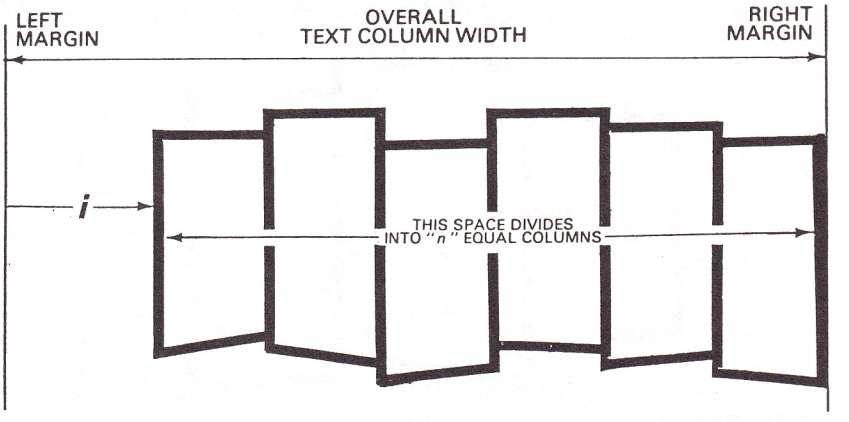
Instead, take the sample above and decrease the number of cols to 4, using the same copy.

(3E) **‹te,4›** + **‹ta›**

C1	C2	C3	C4
27	31	12	5
243	74	38	4
172	44	92	36
167	71	294	

TAB EQUAL Number of Columns





COMMAND automatically computes the widths so that each column will be equal to the others. It does this by first subtracting the indent value, (the first parameter) from the current text column width. This result is then divided by the number of columns wanted. The result is each tab column's width. Once established, the same rules used with the other tab set-ups apply here, except one. The last column **always ends** at the right text margin.

BEGINS when the TAB ACTIVATE **‹ta›** is given. Any line-ending signal causes an advance to the **next column**. The last signal (equal to the number of columns) automatically starts a new line.

ENDS

1. With a TAB CANCEL **‹tx›** command.
2. With a CANCEL ALL Tabs/Indents **‹xx›** command.
3. When another TAB EQUAL **‹te›** command overrides.
4. When a TAB SET, TAB WIDTH, TAB PROPORTIONAL or TAB TEXT set-up is issued.

EXCEPT

1. 20 COLUMNS is the maximum.

CANCEL WITH **TX**

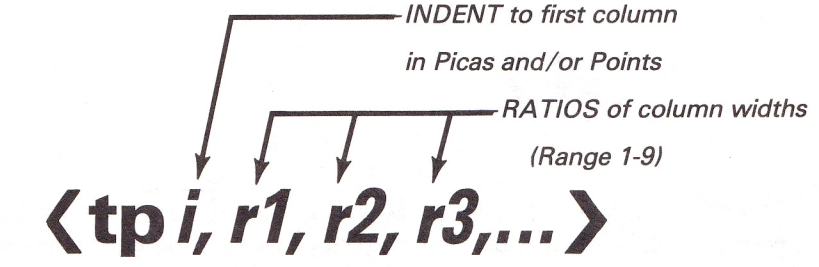
TP•

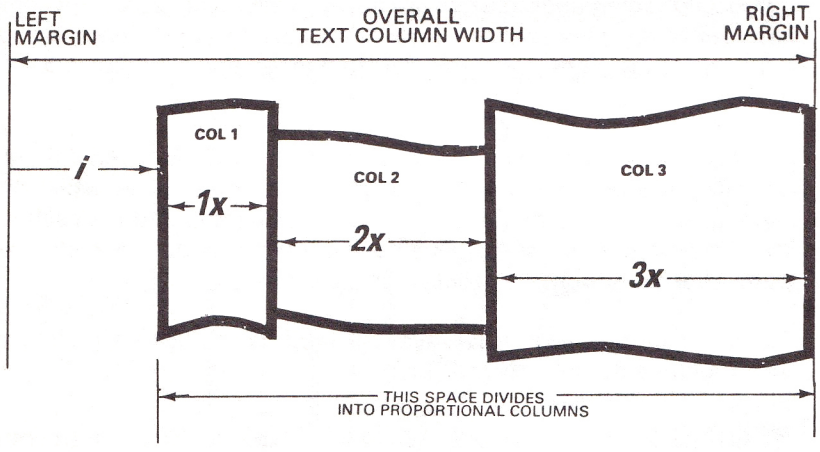
SECT. III—COMPOSITION COMMANDS

3.8 Tabbing Control Group

Tab Proportional

TAB PROPORTIONAL Each Column





COMMAND computes a number of UNEQUAL tab columns given the proportional ratios of each (based on a scale of relative widths of 1 to 9). First the indent value is deducted from the column width. What is left is apportioned according to the ratios given to each column in the command. The last column ends exactly *at* the right margin.

BEGINS when the TAB ACTIVATE **⟨ta⟩** is given. Any line-ending signal causes an advance to the **next column**. The last signal (equal to the number of columns) automatically starts a new line.

ENDS

1. With a TAB CANCEL **⟨tx⟩** command.
2. With a CANCEL ALL Tabs/Indents **⟨xx⟩** command.
3. When another TAB PROPORTIONAL **⟨tp⟩** command overrides.
4. When a TAB STOP, TAB WIDTH, TAB EQUAL or TAB TEXT set-up is issued.

EXCEPT

1. 20 COLUMNS is the maximum.

EXAMPLES of command TP

This command sets up UNEQUAL columns. We'll duplicate (1S) & (1W) once more. But only for a "show how" comparison. To get five equal cols, use any number five times, in this case a 1. However, this command is cumbersome when used for equal cols.

Here is some copy to judge the size of a *Special 12 PICA* column of type before we set up tabs.

(1S) **⟨tp2,1,1,1,1,1⟩ + ⟨ta⟩**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

No indent given? Where is it? You're right. Its distributed into ALL columns.

(2E) **⟨tp,1,1,1,1,1⟩ + ⟨ta⟩**

C1	C2	C3	C4	C5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

The value of this command is for unequal cols. So let's try some. Remember, the parameters are expressed as ratios.

(3E) **⟨tp,7,6,5,4,3⟩ + ⟨ta⟩**

C1	C2	C3	C4	5
27	31	12	5	243
74	38	4	172	44
92	36	167	71	294

Here the 2nd column is twice as wide as the first, the 3rd column is three times the size, etc.

(4E) **⟨tp,1,2,3,4⟩ + ⟨ta⟩**

C1	C2	C3	C4
27	31	12	5
243	74	38	4
172	44	92	36
167	71	294	

TT

Tab Text

CANCEL WITH
TX

EXAMPLES of command TT

This command can be used to set up tabular material using COPY itself to determine column widths.

COPY
Col 1 ‹tt› COLUMN TWO ‹tt› c3 ‹tt› COL 4

```
Col 1 COLUMN TWO c3 COL 4
|   |             |  |   |
  123 123456789012345  1  1234
 4567      6789012345  2 56789
   89        67890123  3   123
```

Even the point size and style can vary, but note below that the right edge of the last column has been automatically set to the current right margin.

```
Col 1 COLUMN TWO c3 COL 4
|   |          |  |    |
  123  4567 890       1234
  456   789  01        234
   56    78   9         10
```

To get the last col exact we put aTab Text after it also, thus:

COPY
Col 1 ‹tt› COLUMN TWO ‹tt› c3 ‹tt› COL 4 ‹tt›

```
Col 1 COLUMN TWO c3 COL 4
|   |          |  |    |
                       |1 -
                        23
  4567  8901234        4567 -
                         89
    01   234  56        78 9
    10   123  45
```

Now that didn't work! Remember, the program ALWAYS goes to the full column width. In this case it made another col (No. 5) which we forgot to take into account with our copy. Here's where we can use a ‹tk› to good advantage to end each line.

```
Col 1 COLUMN TWO c3 COL 4
|   |          |  |    |
  123  4567 890       1234
  456   789  01        234
   56    78   9         10
```

OK! THAT FIXED IT

TAB TEXT Here

(Also called "Tab Pointer")

‹tt›

COMMAND sets Tab Stops anywhere within a line that the command is read. There is also a limit of 20 stops when using this command.

If two ‹tt› commands in a row are issued, with no horizontal movement between them, presumably a zero width column is provided. The program, however, does not permit this to happen and substitutes 6 points of space, instead.

The ‹tt› command can be employed in any type of line, even if there are font and size changes and including justified lines. Lines which have been force justified (ending with a ‹ju› command), lend themselves to the Tab Text form of set-up. One need not worry about character **positions** after expansion. Tab Text takes care of it!

The line measure itself is **always** assumed as the last Tab Stop, and the program automatically assigns it so.

BEGINS later when a TAB ACTIVATE ‹ta› is issued. Remember, YOU must keep track of the number of Tab Text stops which were established, when it comes to actually setting tabular copy.

ENDS

1. With a TAB CANCEL ‹tx› command.
2. With a CANCEL ALL Tabs/Indents ‹xx› command.
3. With the start of new job.
4. When another set-up ‹ts›, ‹tw›, ‹te›, ‹tp› or another ‹tt› arrangement is established.

EXCEPT:

1. If ‹ta› has activated any previous tabular set-up, then ALL ‹tt› 's will be ignored.
2. If a ‹ta› is activated before the line containing Tab Texts is terminated, then the ‹tt› 's will be ignored.
3. If the ‹tt› commands are distributed along a justified line which OVERSETS, then only the ‹tt› points which are in the CARRIED OVER portion that goes to the next line will be established.

CANCEL WITH
TC

TI●

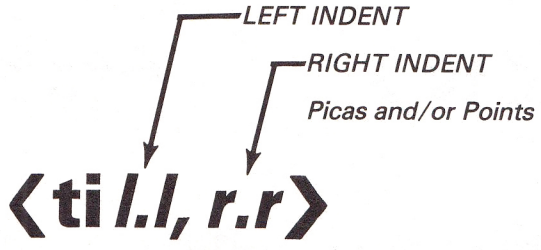
SECT. III—COMPOSITION COMMANDS

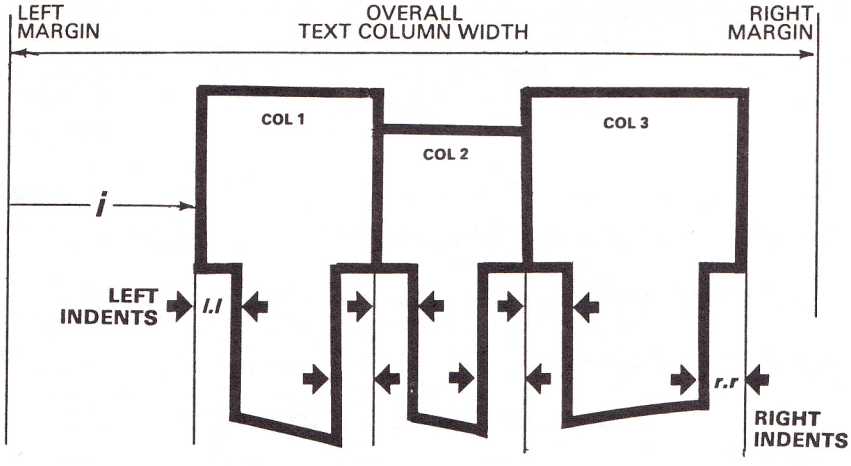
3.8 Tabbing Control Group

Tab Indent

TAB INDENT Each Column

(Also called "Tab Gutter")





COMMAND allows either a left, right or both indent amount to be applied equally to ALL columns,including the last. Use of this command is a simple solution to the "gutter" problem associated with tabs. Either the left or right indent can be set to the "gutter space", but remember that the first column will indent, or the last column fall short, by the same amount, as the case may be.

APPENDS to any of the previous tab set-ups:

1. TAB SET.
2. TAB WIDTH.
3. TAB EQUAL.
4. TAB PROPORTIONAL.

BEGINS when TAB ACTIVATE **⟨ta⟩** is issued to start Tab Set-up going.

ENDS

1. When a TAB INDENT CANCEL **⟨tc⟩** is issued. Note that this only cancels the indent value, not the tab set-up itself.

EXAMPLES of command TI

This is NOT a set-up. It attaches itself to any previous set-up. Below we start off with the now familiar (1S) example. Copy is represented by 3 digits quadded left and the column edges by vertical rules.

Here is some copy to judge the size of a standard 11 pica column of type before we set up tabs.

(1S) **⟨ts2,4,6,8,10,12⟩ + ⟨ta⟩**

Set-up tabs, activate.
No indent, copy quad left

C1	C2	C3	C4	C5
123	456	789	123	456

Six points left indent **⟨ti.6⟩**

123	456	789	123	456

One pica left indent **⟨ti1⟩**

123	456	789	123	456

Six points each side **⟨ti.6,.6⟩**

123	456	789	123	456

One pica right indent ⟨ti,1⟩ copy quad right

123	456	789	123	456

TV

Tab Variable

CANCEL WITH
TX

EXAMPLES of command TV

This is a special tabbing command which sets-up special columns, equal to the white space remaining after all other copy is set. It is similiar to the Indent Hang,(See page 3-47) but operates in tab mode.
Remember the grocery ads for Indent Hang?. No? Then let's show them again right now. Notice that the small type centers between the big type in each example.

STEAK CHOICE N.Y. CUT **2^{29}**

TEA LIPTON'S FLOW-THRU **89¢**

Sugar 5-LBS **73¢**

Now for grocery ads in which the various elements are to be in columns this command lets us set it up without any measuring or numerical tab parameters.
First we set up a line with the WIDEST first and last words.

STEAK ⟨tv⟩ **89¢**

Now we can set lines with copy in the same order it appears,
ie: STEAK quad, NY CUT quad, 2^{29} quad. Notice that all small type now centers in a column of its own, diregarding other copy.

STEAK N.Y. CUT **2^{29}**

TEA LIPTON'S FLOW THRU **89¢**

Sugar 5-LBS **73¢**

TAB VARIABLE Space

(Also called "White Space Column")

⟨tv⟩

COMMAND is actually a shorthand method of writing the following combination of commands:

⟨tt⟩ ⟨qm⟩ ⟨tt⟩
or
⟨tt⟩ ⟨wx⟩ ⟨tt⟩
or
⟨tt⟩ ⟨wr⟩ ⟨tt⟩

Thus, the following situation of copy:

COPY 'A' (Tab Text) (Quad Middle) (Tab Text) COPY 'B'

can be reduced to merely:

COPY 'A' ⟨tv⟩ COPY'B'

Three columns are thus set-up as follows:

1. First column equal to width of COPY 'A'.
2. Second column equal to the white SPACE BETWEEN Copy 'A' and 'B'.
3. Third column equal to width of COPY 'B'.

NOTE: This command can be used several times in the same line. Total white space in the line will be divided equally between each Tab Variable command. The only limitation is that the TOTAL number of columns cannot exceed 20 cols.

BEGINS when TAB ACTIVATE ⟨ta⟩ is issued.

SECT. III—COMPOSITION COMMANDS

3.9 Compose-for-Display Group

ST
Strip Take

STRIP TAKE Composed
(Also called De-Compose, Unjustify)

⟨st⟩

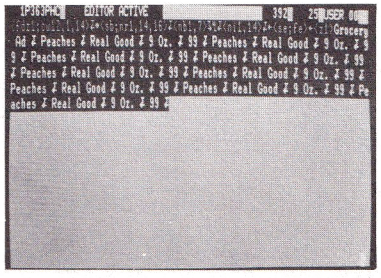
COMMAND has the opposite action of the Compose-for-Display keys. All hyphenation points, line break codes and line information data is "stripped out" from the story file, when passed through the Hendrix Composition Program. You might call it an "UN-Compose". The story reappears on the screen as "originally written" except for type style "color" (ie; dim, reversed, grey backgound etc.), which is not stripped away, but remains on the screen.

IMPORTANT: This command *MUST* be the very first command of the story copy. Insert it at the beginning of the copy line, thus:

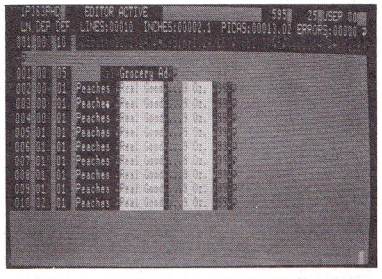
```
   LN  DEP  DEF  LINES:00044  INCHES:00009.9  PICAS:00059.06  ERRORS:00002
⟨st⟩ _____________________________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
_____________________________________________________________
```

EXAMPLES of command ST

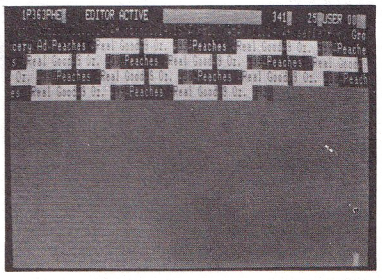
(1) The story shown below appears in its original keyboarded form. Format calls and copy are included.



(2) The Compose-for Display version appears after processing by the HCS program. Bold and italics are indicated as well as spacial relationships. Copy gauge statistics begin each line.



(3) Command STRIP TAKE at the start of the story file has stripped Compose-for-Display and resored the original. Note that Bold and Italic "color" remains.



AE

Allow Estimator

CANCEL WITH

XE

ALLOW ESTIMATOR for Heads

(Also called "Headline Copyfitting")

⟨ae⟩

COMMAND starts the Headline Fit mode, in which special logic is applied to line endings. All lines which follow this command, break at the exact character which exceeds the column width specified. Hyphenation is automatically shut off. The purpose of this command is to aid in exactly copyfitting headlines within allotted news columns. Compose-for-Display gives a preview, with any overset characters vividly showing up as **another line**.

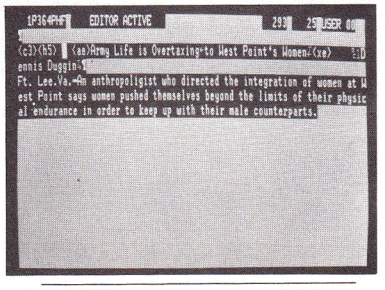
BEGINS immediately. It is commonly included in the Head Formats.

ENDS when a CANCEL ESTIMATOR **⟨xe⟩** is encountered. Hyphenation is automatically restored. (If hyphenation is not wanted, as in ad work, it must be turned off again by a command.)

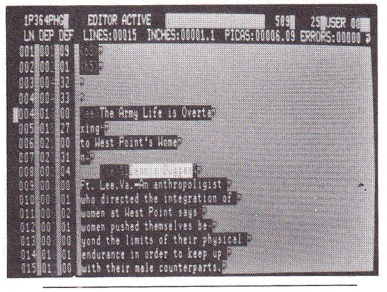
NOTE: The CANCEL ESTIMATOR **⟨xe⟩** command also marks the point of termination for the **Compose-Heads-Only option.**

EXAMPLES of command AE

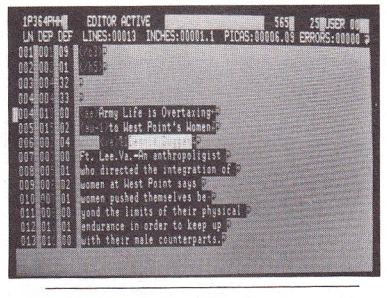
(1) Most often this command is included in a HEADLINE Format. To clarify its use, it's shown immediately before and after the headline.



(2) After Compose-for-Display the screen points out those characters exceeding column width, "spilling" to the next line. "xing" and "n" in this case.



(3) By editing the first line and using Alter Width on the second, the oversize heads now fit.



SECT. III—COMPOSITION COMMANDS

3.9 Compose-for-Display Group

NC
No Count

NO COUNT Line

⟨nc⟩

COMMAND marks those lines which are not to be counted in the total line reported in the classified ad lineage total or in story files. This figure shows up after Compose-for-Display.

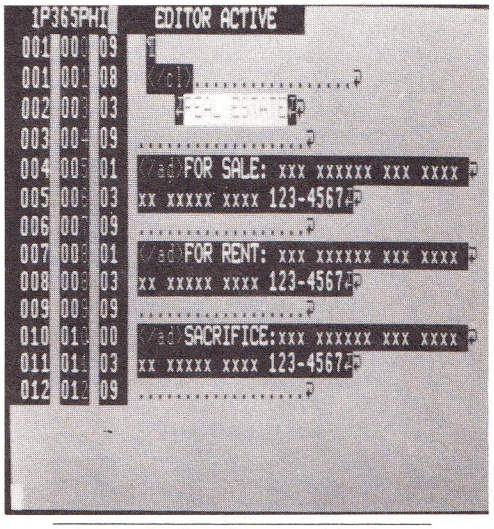
In most newspapers the cut-off rule should not be included in the billing depth. The No Count command acomplishes this.

EFFECTIVE only for the current line on which it is issued.

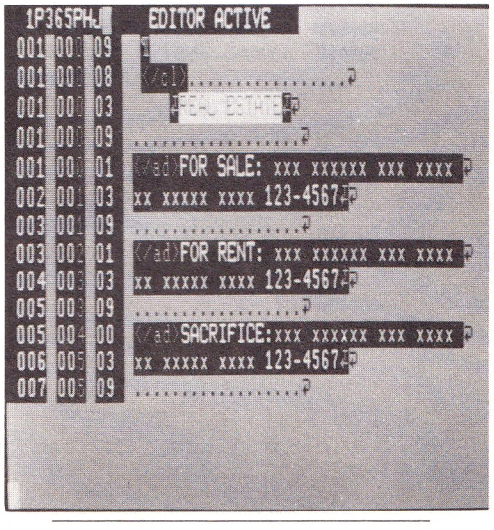
ENDS on the next line.

EXAMPLES of command NC

(1) Here is a portion of classified ads and the formats controlling them. Each line is being counted, including cut-off rules and classification heads, as follows:
Lines 1,2,3 - Classification
Lines 4,5,6 - 1st Ad + cut-off rule.
Lines 7,8,9 - 2nd Ad + cut-off rule.
There are 12 total lines.
Depth is 1.2 inches.



(2) The No Count command has been inserted in the formats, so that cut-off rules and the classification heads will be "skipped" in the accumulated depth. Notice that Line 6 is the sixth actual line of ad copy. Depth now is only .5 inches.



GD•

Galley Define

CANCEL WITH
GX

EXAMPLES of command GD

Below is shown a simplified galley output using the parameters below. BOLD lines indicate the discretionary range
Width - 2.6 pica galley
Depth - 12 picas
Gutter - 6 points
Number - 4 galleys
Discr. Depth -9.9 picas

‹gd 2.6,12,.6,4,9.9›

UNIT ONE

000001	000017	000033	000049
000002	000018	000034	000050
000003	000019	000035	000051
000004	000020	000036	000052
000005	000021	000037	000053
000006	000022	000038	000054
000007	000023	000039	000055
000008	000024	000040	000056
000009	000025	000041	000057
000010	000026	000042	000058
000011	000027	000043	000059
000012	000028	000044	000060
000013	000029	000045	000061
000014	**000030**	**000046**	**000062**
000015	**000031**	**000047**	**000063**
000016	**000032**	**000048**	**000064**

UNIT TWO

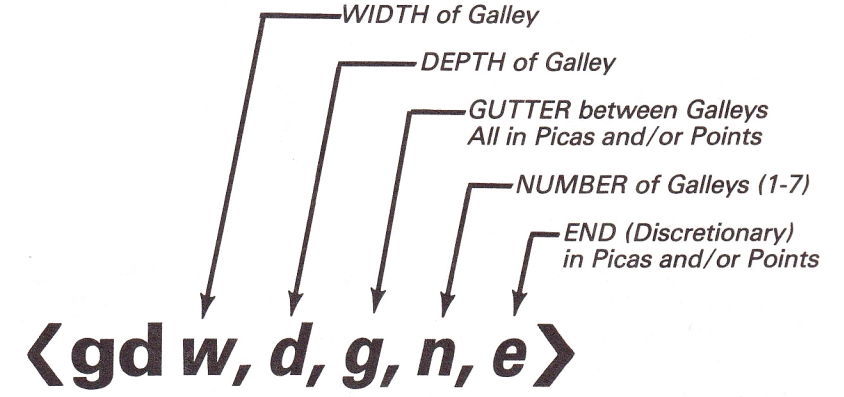
000065	000081	000097	000113
000066	000082	000098	000114
000067	000083	000099	000115
000068	000084	000100	000116
000069	000085	000101	000117
000070	000086	000102	000118
000071	000087	000103	000119
000072	000088	000104	000120
000073	000089	000105	000121
000074	000090	000106	000122
000075	000091	000107	000123
000076	000092	000108	000124
000077	000093	000109	000125
000078	**000094**	**000110**	**000126**
000079	**000095**	**000111**	**000127**
000080	**000096**	**000112**	**000127**

UNIT THREE

000129
000130
000131
000132
000133
000134
000135
000136

GALLEY DEFINE

(Also called "Columnization")



(See diagram on next page.)

COMMAND simultaneously turns on "galley mode" of operation as well as setting up all the paramenters shown in the command and the diagram on the facing page. There are however, certain considerations:

W - GALLEY WIDTH: If it is omitted or zero (0), the program assumes the current column width **‹cw›** value.

D - DEPTH OF GALLEY: This value cannot exceed the reverse lead capacity of the output typesetter.

G - GUTTER: This may be conveniently expressed as points alone. (*ie. -as .15 pts instead of 1.3 picas*).

N - NUMBER OF COLUMNS: Limited to seven (7). If zero is given, the program will calculate the maximum number that will fit in.

E - END, DISCRETIONARY: Use of this parameter is optional. It establishes a threshold depth, useful in classified to keep each ad as an unsplittable unit.

BEGINS as soon as command is read. Program is now in "galley mode".

ENDS with a GALLEY CANCEL **‹gx›** command. The program is placed back into a "line-by-line" mode. The galley in progress is "flushed out" (*See next page for explanation.*)

EXCEPT:

1. **This command works ONLY when phototypesetters are equipped with Reverse Leading Options.**
2. COLUMN WIDTH **‹cw›** command will cancel galley mode completely.
3. GALLEY CANCEL **‹gx›** command will cause an error if issued when program is NOT in Galley Mode.

SECT. III—COMPOSITION COMMANDS

3.9 Compose-for-Display Group

GE
Galley End

GALLEY END Point
(Also called "Stop Galley")

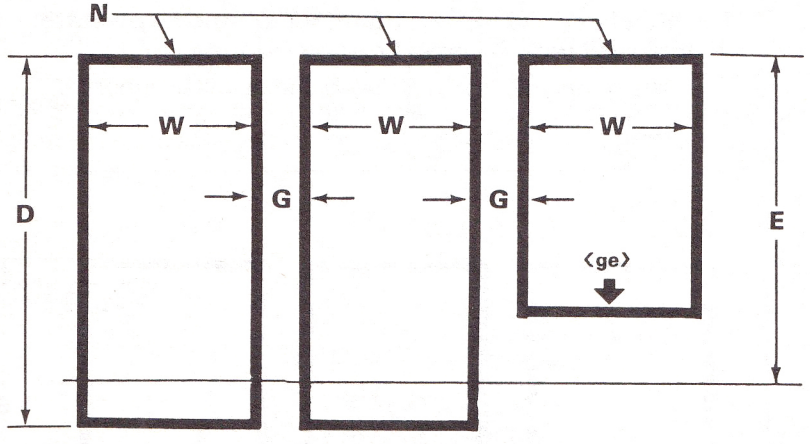
⟨ge⟩

COMMAND forces a galley to end at the point specified instead of going on to the normal depth.

BEGINS on the current line. The command will cause the present galley to end. Copy continues to the next galley.

EXCEPT:

1. **This command works ONLY when phototypesetters are equipped with Reverse Leading Options.**



W - WIDTH of Galley in Picas and/or Points
D - DEPTH of Galley in Picas and/or Points
G - GUTTER between Galleys in Picas and/or Points
N - NUMBER of Columns
E - END, DISCRETIONARY of Galley in Picas and/or Points

'FLUSH OUT' means to stop at the current line (except in the first col) and proceed to a new "unit", by advancing to galley depth, automatically leading 48 points, ending at the first line of a new "unit".
(See line no. 119 in Unit Two at right. The program "flushes out" directly to line 129 in Unit Three.)

EXAMPLES of command GE

The example on the opposite page now appears with Galley End points. The galley mode continues. This command affects only an individual galley.

UNIT ONE

000001	000017	000033	000049
000002	000018	000034	000050
000003	000019	000035	000051
000004	000020	000036	000052
000005	000021	000037	000053
000006	000022	000038	000054
000007	000023	000039	000055
000008	000024	000040	000056
000009	000025	000041	000057
000010	000026	000042	000058
000011	000027	000043	000059
000012	000028	000044	000060
000013	⟨ge⟩	000045	000061
000014		**000046**	**000062**
000015		**⟨ge⟩**	**000063**
000016			**000064**

UNIT TWO

000065	000081	000097	000113
000066	000082	000098	000114
000067	000083	000099	000115
000068	000084	000100	000116
000069	000085	000101	000117
000070	000086	000102	000118
000071	000087	000103	⟨ge⟩
000072	000088	000104	
⟨ge⟩	000089	000105	
	000090	000106	
	⟨ge⟩	000107	
		000108	
		000109	
		000110	
		000111	
		000112	

UNIT THREE

000129	000145
000130	000146
000131	000147
000132	
000133	
000134	
000135	
000136	
000137	
000138	
000139	
000140	
000141	
000142	
000143	
000144	

GF

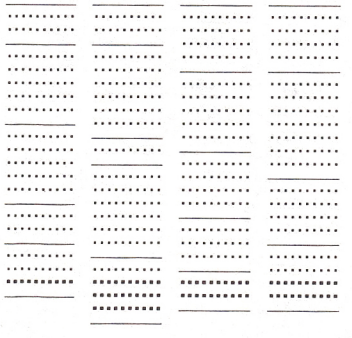
Galley Free

EXAMPLES of command GF

*Several Galley Free commands are shown in the columns below. They will be ignored unless they fall within the"discrectionary range" (between parameters "**d**" and "**e**" of the Galley Define. These lines are shown in bold type. When this happens, the galley ends right then and there.*

000006	000022	000038	000054
000007	‹gf›	000039	000055
000008	000024	000040	‹gf›
000009	000025	000041	000057
000010	000026	000042	000058
000011	000027	‹gf›	000059
‹gf›	000028	000044	‹gf›
000013	000029	000045	000061
000014	**‹gf›**	**000046**	**000062**
000015		**000047**	**‹gf›**
000016		**000048**	

A very practical and common use of this command is in the classified ad pages. Below is a symbolic page. Dotted lines represent copy; solid lines are cut-off rules. To prevent adsfrom being split, a Galley Free command is included with each rule. If it lands in the discretionary range (bold dotted lines), the NEXT ad goes to the NEXT column.



GALLEY FREE Point

(Also called "Galley If Point")

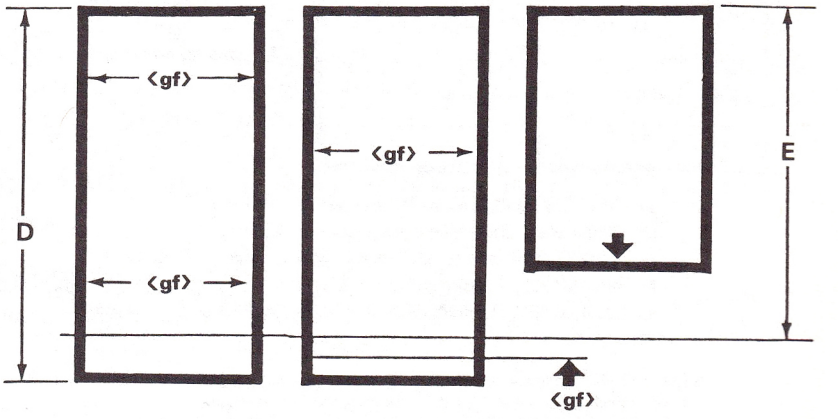
‹gf›

COMMAND is a galley end point with special qualifications and restrictions. This command is ignored, however, unless it is issued between the Galley Discretionary End and the Full Galley Depth (*Parameters "e" and "d" in Galley Define command and the figure below*). Under these conditions it becomes the end of a galley.

BEGINS when issued at the start of a line.

EXCEPT:

1. **This command works ONLY when phototypesetters are equipped with Reverse Leading Options.**
2. If the program is not in Galley Mode, it will be ignored.



ONLY THIS COMMAND
IS HONORED

PC•
Photo Compose

SECT. III—COMPOSITION COMMANDS
3.10 Miscellaneous Command Group

PHOTO COMPOSE Device (& Font) Select

(Also called "Select Output Device")

NUMBER of phototypesetter unit (either 0 or 1)

〈pc *n*〉

COMMAND specifies which of two possible phototypesetters is to be used, (0 for the FIRST typesetter, 1 for the SECOND typesetter). The program then refers to the appropriate "default" values, font width tables, point size tables and other data pertinent to the particular machine in question. It employs this data to process text for presentation on the terminal through the Compose-for-Display program. Thus a simulation on the screen of either output typesetter can be achieved.

BEGINS at the beginning of a story file. The command PHOTO COMPOSE **〈pc〉** *must* be the first command in a story. The only exception to this rule is the Strip Take **〈st〉** command, which precedes even the **〈pc〉** command.

EXCEPT if no **〈pc〉** command is given (at the start of EACH story) the program will issue a default of zero (0) and assume the FIRST of two machines is in use.

OPTION

NUMBER of Phototypsetter

FONT NOS. (a to h represent Machine positions 1 - 8)

〈pc *n,a,b,c,d,e,f,g,h*〉

COMMAND also allows any of the Font Tables to be specified as a "machine arrangement". *(Up to 64 Font Tables may be stored in the 6400 system.)* This optional form is intended for phototypesetters or CRT setters that have individual segments, strips, grids etc. that may be rearranged in an infinite variety of ways.

EXAMPLES of command PC

Assume for the moment that we have two phototypesetters that are alike. Each carries up to eight individual grids loaded in any order.

The following commands are placed at the start of story files. IMPORTANT: They MUST be the first commands given. Let's analyze the ones below.

〈pc0, 1,2,3,4,9,10,11,12〉

Output to the first phototypesetter (Device 0) has been set-up so that:

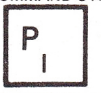
Font Table **1**	*will access*	**1st** Grid Position
Font Table **2**	*will access*	**2nd** Grid Position
Font Table **3**	*will access*	**3rd** Grid Position
Font Table **4**	*will access*	**4th** Grid Position
Font Table **9**	*will access*	**5th** Grid Position
Font Table **10**	*will access*	**6th** Grid Position
Font Table **11**	*will access*	**7th** Grid Position
Font Table **12**	*will access*	**8th** Grid Position

〈pc1, 9,10,47,48,53,54,3,4〉

Output to the second phototypesetter (Device 1) has been set-up so that:

Font Table **9**	*will access*	**1st** Grid Position
Font Table **10**	*will access*	**2nd** Grid Position
Font Table **47**	*will access*	**3rd** Grid Position
Font Table **48**	*will access*	**4th** Grid Position
Font Table **53**	*will access*	**5th** Grid Position
Font Table **54**	*will access*	**6th** Grid Position
Font Table **3**	*will access*	**7th** Grid Position
Font Table **4**	*will access*	**8th** Grid Position

SINGLE KEY COMMAND SYMBOL



PI•

PI Call

3.10 Miscellaneous Command Group

EXAMPLES of command PI

Below are some of the Pi assignments for the particular grid used for this manual. Next, a few uses of Pi symbols are shown.

<pi> f (orF)_____for
<pi> l (orL)_____lbs
<pi> c (orC)_____¢ lb.
<pi> s (orS)_____☆
<pi> r (orR)_____®

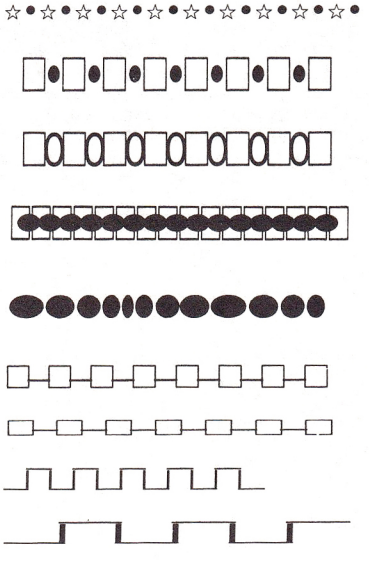
Widgit®

3 FOR 87¢

5 LBS $1.29

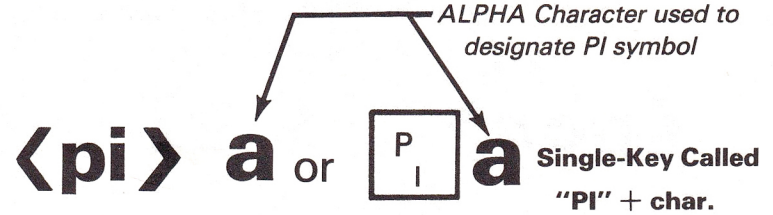
39¢ lb.

Symbols can also be used to create border designs. Designs are limited only by the imagination.



PI CHARACTER (and Font) Call

(Also called "Special Symbol Call")



COMMAND calls out a special character or symbol that is on the grid, filmstrip, disc, etc., but NOT on the keyboard. The "Pi Key" plus an ALPHA calls out these special characters. The particular letters that pick out specific PI symbols have already been specified for your program. Capital letters may stand for different symbols than the lower case equivalent.

IMPORTANT: The program not only finds the symbol itself, but also the proper font or grid number, if it is not on all fonts. It then automatically returns to the font that was in use before the Pi call came along. It is known as a "one-shot" command. Thus if several Pi symbols in a row are needed, Pi Call must be given BEFORE EACH.

EXAMPLES: Those shown below are typical of designated Alphas. Your system is, no doubt, different. Specifications were decided before your system was delivered, and now form part of your "customization" package.

P I A selects A/C sign

P I B selects BOX, Solid

P I b selects BOX, Open

P I C selects CENTS sign ("Big" C)

P I c selects SUPERIOR Cents ("Small" C)

P I R selects RIGHT ARROW (Bold)

P I r selects RIGHT ARROW (Light)

SECT. III—COMPOSITION COMMANDS

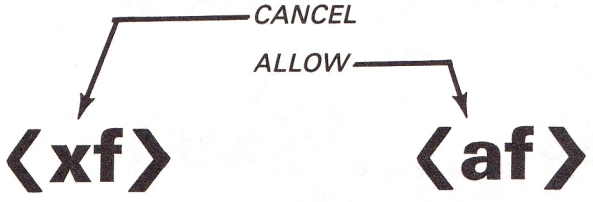
3.10 Miscellaneous Command Group

XF/AF

Cancel Flash/Allow Flash

CANCEL FLASH/ALLOW FLASH

(Also called "Turn On/Off Lamp")



COMMAND inhibits the exposure of all subsequent characters by turning off the Xenon flash lamp (or CRT beam). Escapement of characters continues normally. The effect is to replace each character with a blank space of equal width. This command can be useful when a horizontal movement equal to the size of a word is needed, without the word itself showing.

BEGINS immediately.

ENDS when an ALLOW FLASH **〈af〉** is issued.

EXAMPLES of command XF, AF

(1) First let's establish the copy:

ONE - TWO - THREE - FOUR - FIVE - SIX - SEVEN - EIGHT - NINE - TEN -

(2) Before every EVEN number issue a CANCEL FLASH. Before every ODD number issue an ALLOW FLASH.

ONE - - THREE - - FIVE - - SEVEN - - NINE - -

(3) Before every ODD number issue a CANCEL FLASH. Before every EVEN number issue an ALLOW FLASH.

- TWO - - FOUR - - SIX - - EIGHT - - TEN -

If you have no reverse leading capability, No Flash can help immensly. Follow these steps.
(1) First set-up the following two lines.

PEARS SWEET
PEARS TASTE

Now close up the leading so that 'SWEET TASTE' is correct.

PEARS SWEET
PEARS TASTE

Finally issue a NO Flash before PEARS and an ALLOW FLASH after. Here's the reult.

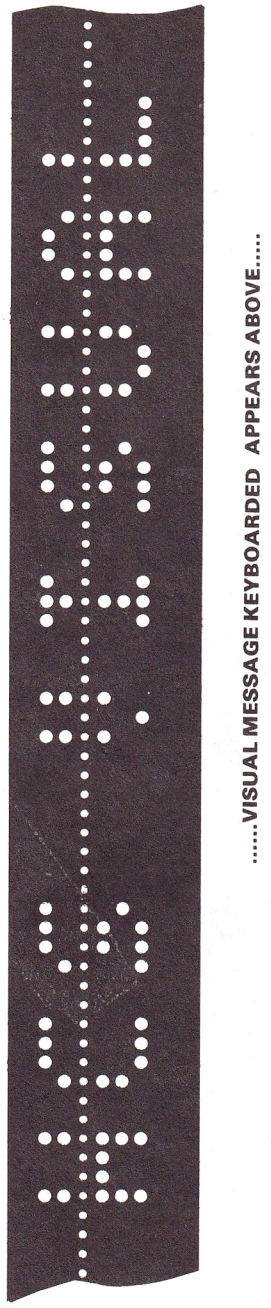
PEARS SWEET TASTE

VI/XV

Visual Start/Visual End

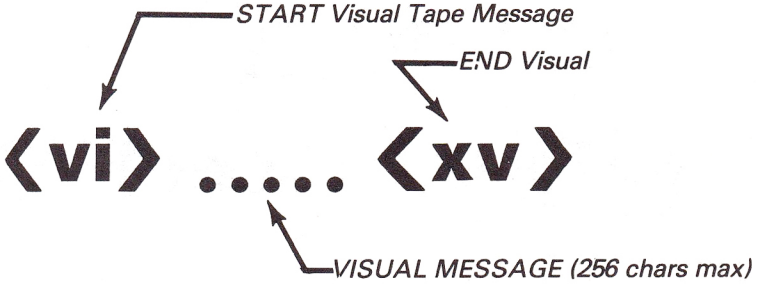
EXAMPLES of command VI

Here is an actual visual produced by using this command.



VISUAL START/END on Tape

(Also called "Tape Identification")



COMMAND instructs the program to output the characters that follow the command **〈vi〉** as "visuals" (punch patterns recognizable as alphanumeric characters), rather than machine coding (such as TTS). The end of the message is signalled by the **〈xv〉** command.

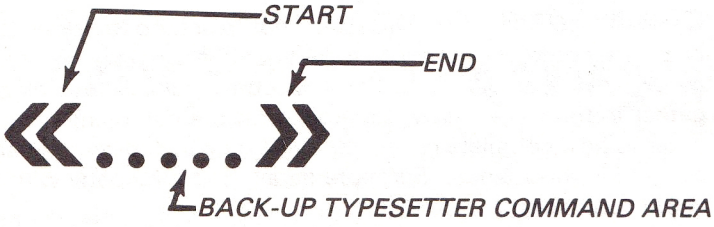
Visuals are a great help in identifying individual tapes that are to be stored away. Visuals can also provide special instructions for running a tape (such as the paper width to be loaded in the typesetter).

BEGINS immediately, if a tape is to be punched. Otherwise, the message is ignored.

ENDS immediately when VISUAL END **〈xv〉** is encountered. Normal coding then follows.

BACK-UP START/END

(Also called "Off-Line Command")



PRECEDENCE CODES above are used to tell the program to "Start (or End) Back-up Mode". The program then allows the coding between them to be sent *directly* to the phototypesetter, if assigned the proper queue line. (*See Item "Except 1" below* .)

All commands MUST be coded in a language that is understood by your typesetter.

When the "End" signal is read, the text following is put out in the normal text coding used, generally, but not necessarily, TTS.

Fundamentally, this is a situation that is used when the 6400 Composition program is inoperative and there is a need to address the typesetter directly from the story files.

NOTE: "Double Bent Brackets" can also be used to add "comments" or perhaps, "instructions" to a job. That is, copy between them will appear on the screen at all times, BUT will be "stripped" when output normally (through HCS) to the typesetter. (*See Item "EXCEPT 2" below.*)

BEGINS when BACK-UP START « is read.

ENDS when BACK-UP END » is read.

EXCEPT:

1. The BACK-UP START/END **recognizes** commands only when an output queue is selected that goes directly to the typesetter, thus *bypassing* the HCS program.
2. The BACK-UP START/END **ignores** commands when outputs to the typesetter (or terminals) are *processed* by the HCS program.
3. The TYPESETTER COMMAND itself cannot be more than 256 characters. It is automatically terminated at 256 characters, if it is greater. If such a problem arises, then employ an EXTRA PAIR of Start/End signals and continue the command there.

CANCEL

CANCEL COMMANDS

(Also called "Reset", "Default")

COMMANDS which can cancel out, reset, go to "*default*" condition or in some other way override a command in progress, are known by the generic name of CANCEL. For your convenience they are grouped together in the table below, together with the commands they affect. In order to be a complete reference, this listing includes commands not yet covered in this manual. For more detail, go to the page with the uncancelled command.

Cancel Format	**FX**	*cancels*	**FD—FE**	**Format in Progress**[1]
Cancel Galley	**GX**	*cancels*	**GD**	**Galley Mode**
Cancel Hang	**HX**	*cancels*	**HI**	**Hang Indent**
"	**HX**	*cancels*	**HH**	**Hang Here**
Indent Hang Reset	**IR**	*cancels*	**IH**	**Indent Hang**
Cancel Indent Take	**IX**	*cancels*	**IT**	**Indent Take**
Cancel Length	**LX**	*cancels*	**LL**	**Length Left**
"	**LX**	*cancels*	**LR**	**Length Right**
Positive Video	**PV**	*cancels*	**NV**	**Negative Video**
Quad Paragraph	**QP**	*cancels*	**IP**	**Indent Paragraph**
Cancel Ragged	**RX**	*cancels*	**RL**	**Ragged Left**
"	**RX**	*cancels*	**RC**	**Ragged Center**
"	**RX**	*cancels*	**RR**	**Ragged Right**
Cancel String	**SC**	*cancels*	**SB—SE**	**String in Progress**
Set Upright	**SU**	*cancels*	**SO**	**Set Oblique**
Cancel Skew	**SX**	*cancels*	**SL**	**Skew Left**
"	**SX**	*cancels*	**SR**	**Skew Right**
Cancel Tab Indent	**TC**	*cancels*	**TI**	**Tab Indent**
Cancel Tabs	**TX**	*cancels*	**TS**	**Tab Set**
"	**TX**	*cancels*	**TB**	**Tab Width**
"	**TX**	*cancels*	**TE**	**Tab Equal**
"	**TX**	*cancels*	**TP**	**Tab Proportional**
"	**TX**	*cancels*	**TI**	**Tab Indent**
"	**TX**	*cancels*	**TT**	**Tab Text**
Visual End	**XV**	*cancels*	**VI**	**VIsual String**
Cancel Auto	**XA**	*cancels*	**AA**	**Allow Auto**
Cancel Dictionary	**XD**	*cancels*	**AD**	**Allow Dictionary**
Cancel Estimator	**XE**	*cancels*	**AE**	**Allow Estimator**
Cancel Flash	**XF**	*cancels*	**AF**	**Allow Flash**
Cancel Hyphenation	**XH**	*cancels*	**AH**	**Allow Hyphenation**
Cancel Letterspace	**XL**	*cancels*	**AL**	**Allow Letterspace**
Cancel MAX Band	**XM**	*cancels*	**AM**	**Allow MAX Band**
Cancel MIN Band	**XN**	*cancels*	**AN**	**Allow MIN Band**
Cancel ALL Tab/Indent	**XX**	*cancels*	**IT**	**Indent Take**
"	**XX**	*cancels*	**HI**	**Hang Indent**
"	**XX**	*cancels*	**HH**	**Hang Here**
"	**XX**	*cancels*	**IH**	**Indent Hang**
"	**XX**	*cancels*	**LL**	**Length Left**
"	**XX**	*cancels*	**LR**	**Length Right**
"	**XX**	*cancels*	**TA**	**Tab Activate**
"	**XX**	*cancels*	**TI**	**Tab Indent**

[1]**NOTE:** ⟨FX⟩ command also reestablishes the HSETUP "default" values, namely ⟨cw⟩, ⟨fn⟩, ⟨ps⟩, ⟨vl⟩, then calls ⟨xx⟩.

SECT. III—COMPOSITION COMMANDS

Compugraphic Videosetter
UNIVERSAL NV/PV

3.12 Special Typesetter Model Commands

Negative Video/Positive Video

NEGATIVE/POSITIVE VIDEO

(Aso called Reversed Type, White-on-Black)

〈nv〉 **〈pv〉**

COMMANDS allow a reversal of the black-&-white "color" characterisitics.

NEGATIVE VIDEO produces white type on a black background.

POSITIVE VIDEO produces black type on a white background.

BEGINS immediately when command is read.

ENDS when the counterpart command (either **〈pv〉** or **〈nv〉**) is encountered.

EXCEPT

1. If leading more than the point size is used during negative video, the extra amount will appear as white, not black, space.
2. **A wiring change in the typesetter is necessary to accomodate these command. Consult the manufacturer.**

EXAMPLES of command NV, PV

The lines below are set solid (12/12) using Negative Video.

This is Negative Video
This is Negative Video
This is Negative Video

If 4 points of leading is added, a white space appears between the lines.

This is Negative Video

This is Negative Video

This is Negative Video

There is a way to fix this problem. Using a format, set a line (use Fill with Space) using 5 point type, (also reduce copy leading by 5 points to 11, (ie; 12+4—5). Fixed spaces were added before and after the copy to enlarge the black area.

This is Negative Video
This is Negative Video
This is Negative Video

Here's an actual sample from a newspaper.

10th Ave So 2753. 4BR. lge kitch, LR & DR. Newly decor nat wd flrs & trim. Bdrms cptd. Must see to appreciate. 10 min to U of M $290 pl util 929,4418, 871-3942.

Apts. Unfurnished

Atten. students: 1811 Park--eff. w/kit $95; 2 BR w/kit $150 735-3633 or 222-3601.

Clean 1 BR w/AC, stove, frig, heat Pleasant & Lake 926-5996.

Eff. apt 53rd & Lyn. S. 1 blk to U exp. bus avbl 3/1 $125 822-7413.

Eff partly furn. $135/mo incl util. Exc elec. Pets OK. Call 331-7588.

Fonts

Compugraphic Videosetter
UNIVERSAL

3.12 Special Typesetter Model Commands

GENERAL NOTES:

The Videosetter Universal will accept 8 grids (4 in each turret), in any order. They are placed into "Grid Positions 1-8". This is known as a "Set-Up".

The Universal automatically loads width values from optical binary coding contained on each grid. Eight corresponding Font Width Tables are in the 6400 HCS program under a file called "HFONTS". HCS has provisions for 64 such tables, thus '8 Set-Ups of 8 Grids' is possible.

Furthermore, with the Videosetter's ability to make "pseudo" type fonts by slanting (Italics), squeezing (Condensed) and stretching (Expanded), there are 12 variations from each grid. Using all 8 set-ups yields an incredible total of **768 Fonts** which need identifying numbers! This full range is depicted in the column at the left, showing the eight set-ups. Obviously, the FONT NUMBER **〈fn *n*〉** command is inadequate when there are so many *n* 's to contend with. The solution to this problem is:

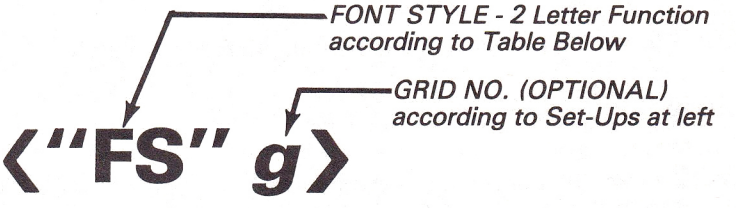
8 SET-UPS

The 8 SET-UPS containing 8 Grids in each will contain the following Grid and Font Numbers.

SET-UP	FONT NUMBER	GRID
1	1-12	1
	13-24	2
	25-36	3
	37-48	4
	49-60	5
	61-72	6
	73-84	7
	85-96	8
2	97-108	9
	109-120	10
	121-131	11
	132-144	12
	145-156	13
	157-168	14
	169-180	15
	181-192	16
3	193-205	17
	206-217	18
	218-228	19
	229-240	20
	241-252	21
	253-264	22
	265-276	23
	277-298	24
4	289-301	25 to 32
	373-384	
5	385-393	33 to 40
	469-480	
6	481-493	41 to 48
	565-576	
7	577-588	49 to 56
	661-672	
8	673-684	57 to 64
	757-768	

FONT STYLE/GRID CHANGE

(Also called "Videosetter Type Change")



COMMAND is actually twelve commands using the 2-letter function mnemonics listed below. This secures the appropriate "pseudo" font, with the option of selecting a grid change simultaneously. If followed by commas, the point size **〈ps〉** and vertical lead **〈vl〉** may also be included, so that;

〈ci*6, 24, 30*〉

Change to: Condensed Italic on Grid 6
24 pt on 30 pt lead

NOTE: This command will automatically pick up the *"default"* values from individual Font Width Tables. They are: (1) Minimum Spaceband, (2) Maximum Spaceband, (3) Condensing Value, and (4) Expansion Value, in percentage.

BEGINS immediately when command is read.

ENDS with next valid FONT NO. or FONT STYLE commands.

TWELVE "PSEUDO" FONT STYLES FROM ONE GRID

STYLE NO.	TWO LETTER FUNCTION	COMMAND DESCRIPTION	COMMAND & PARAMETERS
1	NR•	NORMAL Width ROMAN	〈nrg〉
2	NI•	NORMAL Width ITALIC	〈nig〉
3	NB•	NORMAL Width BOLD	〈nbg〉
4	ND•	NORMAL Width BOLD ITALIC	〈ndg〉
5	CR•	CONDENSED Width ROMAN	〈crg〉
6	CI•	CONDENSED Width ITALIC	〈cig〉
7	CB•	CONDENSED Width BOLD	〈cbg〉
8	CD•	CONDENSED Width BOLD ITALIC	〈cdg〉
9	ER•	EXPANDED Width ROMAN	〈erg〉
10	EI•	EXPANDED Width ITALIC	〈eig〉
11	EB•	EXPANDED Width BOLD	〈ebg〉
12	ED•	EXPANDED Width BOLD ITALIC	〈edg〉

Compugraphic Videosetter
UNIVERSAL Fonts

SECT. III—COMPOSITION COMMANDS

3.12 Special Typesetter Model Commands

VIDEOSETTER UNIVERSAL - SAMPLE GRID SET-UP NO. 1

GRID ONE
News No. 2

	CMMND CODE	FONT NO
EXAMPLE of	nr1	1
EXAMPLE of	*ni1*	2
EXAMPLE of	nb1	3
EXAMPLE of	*nd1*	4
EXAMPLE of	cr1	5
EXAMPLE of	*ci1*	6
EXAMPLE of	cb1	7
EXAMPLE of	*cd1*	8
EXAMPLE of	er1	9
EXAMPLE of	*ei1*	10
EXAMPLE of	eb1	11
EXAMPLE of	*ed1*	12

GRID TWO
Univers 55 (Light)

	CMMND CODE	FONT NO
EXAMPLE of	nr2	13
EXAMPLE of	*ni2*	14
EXAMPLE of	nb2	15
EXAMPLE of	*nd2*	16
EXAMPLE of	cr2	17
EXAMPLE of	*ci2*	18
EXAMPLE of	cb2	19
EXAMPLE of	*cd2*	20
EXAMPLE of	er2	22
EXAMPLE of	*ei2*	22
EXAMPLE of	eb2	23
EXAMPLE of	*ed2*	24

GRID THREE
Futura Medium

	CMMND CODE	FONT NO
EXAMPLE of	nr3	25
EXAMPLE of	*ni3*	26
EXAMPLE of	nb3	27
EXAMPLE of	*nd3*	28
EXAMPLE of	cr3	29
EXAMPLE of	*ci3*	30
EXAMPLE of	cb3	31
EXAMPLE of	*cd3*	32
EXAMPLE of	er3	33
EXAMPLE of	*ei3*	34
EXAMPLE of	eb3	35
EXAMPLE of	*ed3*	36

GRID FOUR
Helios

	CMMND CODE	FONT NO
EXAMPLE of	nr4	37
EXAMPLE of	*ni4*	38
EXAMPLE of	nb4	39
EXAMPLE of	*nd4*	40
EXAMPLE of	cr4	41
EXAMPLE of	*ci4*	42
EXAMPLE of	cb4	43
EXAMPLE of	*cd4*	44
EXAMPLE of	er4	45
EXAMPLE of	*ei4*	46
EXAMPLE of	eb4	47
EXAMPLE of	*ed4*	48

GRID FIVE
Sans No. 1

	CMMND CODE	FONT NO
EXAMPLE of	nr5	49
EXAMPLE of	*ni5*	50
EXAMPLE of	nb5	51
EXAMPLE of	*nd5*	52
EXAMPLE of	cr5	53
EXAMPLE of	*ci5*	54
EXAMPLE of	cb5	55
EXAMPLE of	*cd5*	56
EXAMPLE of	er5	57
EXAMPLE of	*ei5*	58
EXAMPLE of	eb5	59
EXAMPLE of	*ed5*	60

GRID SIX
Univers 75 (X Bold)

	CMMND CODE	FONT NO
EXAMPLE of	**nr6**	**61**
EXAMPLE of	***ni6***	**62**
EXAMPLE of	**nb6**	**63**
EXAMPLE of	***nd6***	**64**
EXAMPLE of	**cr6**	**65**
EXAMPLE of	***ci6***	**66**
EXAMPLE of	**cb6**	**67**
EXAMPLE of	***cd6***	**68**
EXAMPLE of	**er6**	**69**
EXAMPLE of	***ei6***	**70**
EXAMPLE of	**eb6**	**71**
EXAMPLE of	***ed6***	**72**

GRID SEVEN
Futura Bold

	CMMND CODE	FONT NO
EXAMPLE of	**nr7**	**73**
EXAMPLE of	***ni7***	**74**
EXAMPLE of	**nb7**	**75**
EXAMPLE of	***nd7***	**76**
EXAMPLE of	**cr7**	**77**
EXAMPLE of	***ci7***	**78**
EXAMPLE of	**cb7**	**79**
EXAMPLE of	***cd7***	**80**
EXAMPLE of	**er7**	**81**
EXAMPLE of	***ei7***	**82**
EXAMPLE of	**eb7**	**83**
EXAMPLE of	***ed7***	**84**

GRID EIGHT
Helios Bold

	CMMND CODE	FONT NO
EXAMPLE of	**nr8**	**85**
EXAMPLE of	***ni8***	**86**
EXAMPLE of	**nb8**	**87**
EXAMPLE of	***nd8***	**88**
EXAMPLE of	**cr8**	**89**
EXAMPLE of	***ci8***	**90**
EXAMPLE of	**cb8**	**91**
EXAMPLE of	***cd8***	**92**
EXAMPLE of	**er8**	**93**
EXAMPLE of	***ei8***	**94**
EXAMPLE of	**eb8**	**95**
EXAMPLE of	***ed8***	**96**

EXAMPLES of command FONTS

This is typical of an 8-grid set-up. The command coding would be similar for any other set-up, only the grid numbers would change. The program can store up to 64 Font Width Tables, and therefore 64 Grid Nos. are possible. (Actual turret positions can be assigned in any of the Font Width Tables.)

SO/XO

Set Oblique/Cancel Oblique

SET OBLIQUE/CANCEL OBLIQUE

(Also called "Psuedo Italic", "Slant")

ANGLE OF SLANT: Degrees

⟨so *a*⟩ **⟨xo⟩**

COMMAND changes typeface to a "psuedo" italic by slanting the vertical stroking of the typesetter's CRT tube. If a parameter value is given to "*a* ", then the angle of the slant can be controlled. If the parameter value is omitted, then a standard angle (usually 12½ to 15 degrees) results.

BEGINS immediately, when the command is met.

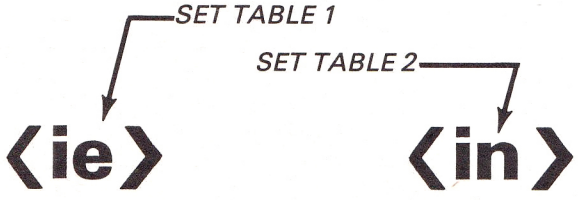
ENDS

1. With a CANCEL OBLIQUE **⟨xo⟩** command.
2. With any command which changes the type font.
3. At the end of a job.

EXCEPT the angle, if given, must be repeated each time the **⟨so⟩** command is used. If a permanent angle change is wanted, it must be made in the "default" values.

INITIALIZE EIGHT/INITIALIZE NINE

(Also called " Set Lens Initialization Tables")



COMMANDS set-up one of the two Lens Initialization Tables that are part of the Pacesetter program. They contain adjustment values to be sure that all lines of copy, regardless of type size will align at the left-hand margin.

INITIALIZE EIGHT command **〈ie〉** is the equivalent of the Pacesetter command "Bell 8" and converts to "Set Table 1", which covers lens sizes from 9 to 72 points.

INITIALIZE NINE command **〈in〉** is the same as a "Bell 9" in Pacesetter coding. It becomes "Set Table 2", which includes the lens sizes smaller than 9 point and up to 72 point.

NOTE: Values in tables vary from machine to machine, depending upon the position and lens sizes in the turret.

BEGINS immediately.

ENDS with the alternative command, Initialize Nine replaces Table 1 with Table 2 and vice versa.

EXCEPT if no command is given the program sets-up "default" values as follows:

1. The *"file name"* which is automatically typeset on the galley at the beginning of each job, (ie. the 'slug line') is always set using command 〈in〉 .
2. The *"story copy"* is set at 〈ie〉 for type for a 9 pt story.

SECTION IV
FORMAT & MACRO COMMANDS

4.1 Introduction

Up to this point, you have learned the meaning of individual composition commands. Each was shown to have a special meaning and a special action.

The real power and versatility of this command "language" is realized when the formatting features of the 6400 Hendrix Composition System (HCS) are fully utilized.

As a mark-up, person you will be dealing in many ways with formats. And macros. Also general variables. But most of the time, with merges.

Well, what are they? Which is used when? And where? And how? A full explanation with examples and illustrations requires a section of this manual unto itself. Initially, it is most important that you learn the proper techniques used to handle and manipulate formats and macros. Hence, this section, deals **only** with the operating commands pertaining to formats, macros, and general variables. Until you are thoroughly familiar with the commands in this section, errors are most apt to be mistakes of omission or of carelessness.

There are three basic steps required before you can enjoy the benefits that formats or macros bring, namely:

1. **DEFINITION**
2. **STORAGE**
3. **RETRIEVAL**

NOTE: The procedure for actually storing formats permanently in the system, so that they are available to **all** users, is described in the 6400 System Operators Manual.

4.2 What is a Format?

A format is a *"string"* of commands and/or characters, identified by a *"name"*, that is *"memorized"* by the computer program, *"stored"* on a disc and can later be *"recalled"*. Formats lend themselves to repeating text passages (ie, standard copy in legals) or constant typographic arrangements (such as headlines, grocery-ad blocks, box-score tab set-ups). Formats are versatile; they can be recycled, interrupted, modified, linked, and may contain mini-formats (macros).

Every format has two essential parts, consisting of:

1. **Name** - from one (1) to six (6) characters long.
2. **Contents** - limited to 124 characters.

NOTE: All formats are "stored", either permanently in the system or on a job-only basis. The word "Format" itself, thus **always** implies that it is a "Stored Format".

Formats may be permanently stored in the 6400 system and hence, they are available to all terminal users at all times. From a mark-up point of view, you merely indicate a call-out at the appropriate spot in the copy. In some cases, only merge points are required to further access formats.

On a job basis, formats need to be written (defined) for the job file before they can be used. In any event, you will need to know how to write and store new formats or read and interpret existing ones. Proficiency in formats, macros etc. is the key to smooth-running, organized, newspaper typography.

4.3 What is a Macro?

Fundamentally, a macro functions in the same manner as a format. There are, however certain physical differences, such as a single digit name and half the capacity. Where they are NOT similar is when it comes to their "hierarchy". Stated simply, this means:

A format CAN contain a macro define.
A macro CANNOT contain a format define.

Unlike formats, there are no "permanent" macros in the system. Up to ten macros will "belong" either to a format or to a story. Macros, are therefore, in a sense "private". The same ten easily-remembered identifying numbers are thus used over and over, in different formats, in different situations, in different jobs. Each time they produce a different and individualized result.

4.4 What is a General Variable?

A general variable is a method of substitution. Up to eight characters are represented by a single letter from A to Z. Its distinction lies in the fact that it is called from *within* a command or from *within* a format. Neither a macro or a format can do that!

Thus, the great value of a general variable is that by redefining its contents, the general variable, in turn, can alter parameters or replace commands themselves which are buried in formats. This, eliminates the need to rewrite a format or use a lengthy command string in its place.

4.5 What is a Merge?

Merges are simply "interruptions" that are under control. We interrupt a format and transfer to the text. Then we interrupt the text and return to the format. Or, a macro is interrupted and the program goes to the text. The text in turn is interrupted in order to go back to the macro. These are simple maneuvers to mark-up and to keyboard (only one stroke) but ever so powerful in what they can accomplish.

FD•/FE

Format Define/Format End

EXAMPLES of command FD...FE

First, let's consider the naming of a format. The following are valid names:

A1 or a1
news3
4COL.
Box2.4
Rule$2

These below are NOT because they contain illegal spaces, hyphens, commas, dashes, etc.

A—1 or a-1
news,3
4 COL
Box2—6
Rule: 2

The following format contains a basic command for a news column of type. Note that although FD and FE are shown in caps, this is not necessary but done here for the sake of clarity.

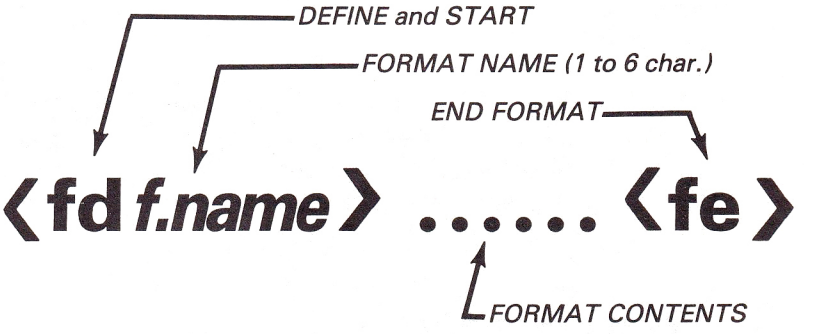
〈FD/news〉 〈cw11,1,9,9.5〉 〈FE〉

〈FD/news ; cw11,1,9,9.5 ; FE〉

In the second version above it has been simplified and shortened by substituting a semicolon (;) for each pair of end-begin symbols 〉 〈.
This can be very helpfull when approaching the 124 character limit of a format.

FORMAT DEFINE/FORMAT END

(Also called "Store Format")



COMMANDS define, start and end a format. The FORMAT DEFINE command **〈fd〉** must also include a "name", which the program uses to identify and locate the format. There are certain rules to follow when choosing a name. There are no restrictions on the format contents. Anything that is proper input to the composition program can be used.

FORMAT NAME must adhere to these restrictions:

1. From one (1) to six (6) characters in length.
2. A *"character"* is a cap or lower case *letter or number, a period or a dollar sign.* No other symbol, character or space is allowed.
3. The program does "force capitals", which means it changes all alpha characters to upper case before storing or searching. This does **not** restrict you from using lower case in defining or calling formats.

FORMAT CONTENTS should consider that:

1. The total number of characters (including the open and close commands and the format name) for any single format must not exceed 124. An error message is output if this limit is passed.

FORMAT STORAGE employs two methods for saving formats which have been properly defined. The first is intended for short-term job-oriented or "private" usage. The second is system-oriented and "public", therefore, accessible to all users.

1. **JOB FORMATS (Temporary):** Up to eight (8) Formats may be stored with each story file. The storing action itself takes place during Compose-for-Display or Compose-for-Output to a

CANCEL WITH **FX**

/f.name

Format Call

(continued from previous page)
typesetter. Formats may be stored at **any** spot, even in the middle of a word, if such a need ever arises. It is common practice to collect all formats, for a particular job, at the beginning. This eases checking, executing, modifying or transferring to a "permanent status".

2. **SYSTEM FORMATS (Permanent):** Up to 240 formats may be stored on a single-disc system (more are optional with an Extended Data Base system). It requires a special technique to store formats permanently. This operation is normally done by the Computer System Operator. Consult him for details. Once stored in this manner, the Formats are available to **all** users of the system.

EXCEPT that:

1. Errors within the commands themselves **are not** detected during storage, but **are** caught when an attempt is made to use the format.
2. Oversized contents cause an error message to appear.
3. Duplicate format names pass through when stored as JOB FORMATS, but are rejected and flagged with an error message when a SYSTEM FORMAT storage loading is done.
4. The command ❮FX❯ in addition to stopping any further action of the format, also reestablishes the HSETUP "default" values, namely ❮cw❯, ❮fn❯, ❮ps❯, ❮vl❯, then calls ❮xx❯, which cancels all Tabs/Indents.

FORMAT CALL

(Also called "Format Activate")

CALL FORMAT (Use "Slash")

NAME (use same name as in "Define")

❮/ f.name❯

COMMAND locates, calls, and activates any format which has been stored. While it is in use, it is known as an "active" format.

BEGINS as soon as FORMAT CALL ❮**/f.name**❯ is read.

EXAMPLES of command FMT CALL

To use the format called "news" on the opposite page it would preceed copy. Let's add another called "hang" and see what happens.

COPY:
❮**/news**❯ The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

❮**/hang**❯ The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

RESULT:
The paragraph you are now reading shows how the Henrdix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

MD• **VALID ONLY FOR OCR INPUT**

SINGLE KEY COMMAND SYMBOL

EXAMPLES of command MD...MD

This is similiar to a format define. Let's show two, using a • to indicate the single key command for Macro Define (acts as both begin and end). Leader dots are for clarity only.

•1.. ⟨fn2⟩ Staff Writer ⟨qc⟩ ..•

•2.. ⟨fn2⟩ Sports Writer ⟨qc⟩ ..•

Normally, this type of macro would be included in a format, so (with some improvements) here they are imbedded in formats:

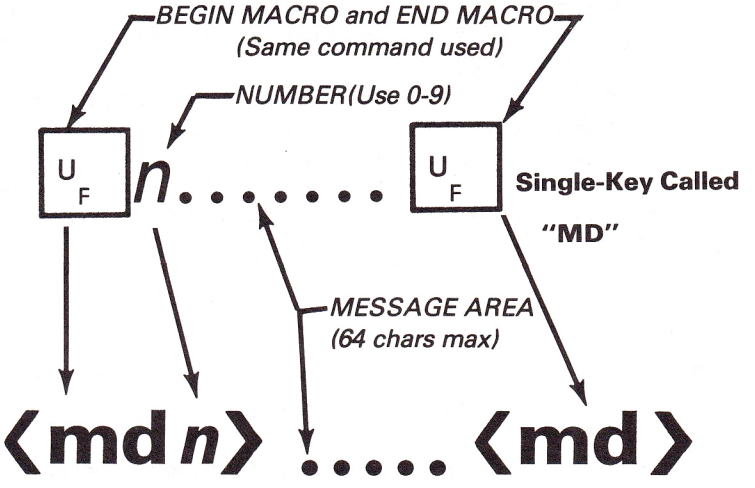
FMT 'news2'

⟨FDnews2⟩ ...
•1 ⟨fn61⟩ Staff Writer
⟨qc;va3;fn1⟩ •...
•2 ⟨fn61⟩ Sports Writer
⟨qc;va3;fn1⟩ •....
(format
itself goes here)
..... ⟨FE⟩

Now when we CALL for the format 'news2', we STORE the macros 1 & 2.
Later we may use both, either or neither, as we wish.
(See opposite page).

MACRO DEFINE

(Also called "Store Macro")



(2-letter command valid only for OCR input)

COMMAND defines, starts, and ends macro. The technique used to establish a macro is similar to that of a format, but there **are** some important differences, as described below:

MACRO NAME: Limited to a single digit (0-9). Obviously then, there can be no more then ten macros existant at one time.

MACRO CONTENTS: Limited to a maximum of 64 characters, comprising commands and/or text. An error message will be output if this maximum is exceeded.

MACRO STORAGE (Temporary): Macros can be stored by any user and at anytime (including from within a format). They are temporary, since they are easily overwritten.

It is a common and useful practice to place macro definitions inside formats. Then, each time the format is activated, all the macros within it are loaded and stored, replacing any that had the same identifying number. In this manner, any format can have its own "private set" of macros to go with it. This feature offers tremendous flexibility.

EXCEPT

1. **Two-letter function commands for MACRO DEFINE are available only for OCR input.** Use single-key commands with system terminal keyboards.
2. Any command errors made within a macro will **not** be detected during storage. They will be caught when an attempt is made to activate the macro with a call.

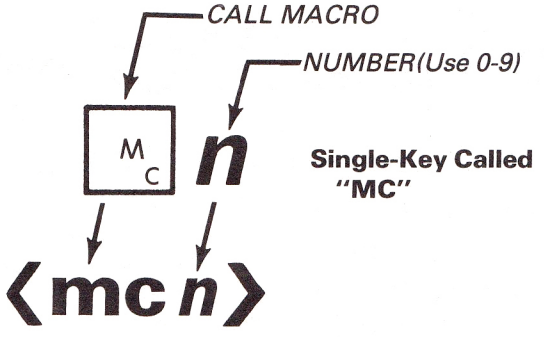
SINGLE KEY COMMAND SYMBOL

VALID ONLY FOR OCR INPUT **MC•**

4.7 Macro Commands — Macro Call

MACRO CALL

(Also called "Get Macro")



(2-letter command valid only for OCR input)

COMMAND (with just two keystrokes) locates call-outs and activates any macro previously stored and identified by the same number.

NOTE: Macros can, and often are, defined within formats or even other macros. Until the format itself is called, and the section containing the macro is read, the macro has not yet been stored away and can not be found by the program. Even worse, an unwanted but previously stored macro, with the same call number, will be activated instead.

BEGINS as soon as command is read.

EXCEPT

1. **Two-letter function command for MACRO CALL is available only for OCR input.**
   Use single-key commands when using system terminal keyboards.
2. Error messages will NOW appear if invalid commands are contained in the macros.

EXAMPLES of command MACRO CALL

Using the same examples shown under 'format call'on page 4-5 lets add the macro calls just defined within the format. First we define the macros by calling out "/news2". This needs to be done only once. Better still would be to have "/news" automatically call "/news2". That way you will never forget!

COPY:
⟨/news2⟩
⟨/news⟩ •1The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

⟨/hang⟩ •2The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

RESULT:

Staff Writer

The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

Sports Writer

The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

SINGLE KEY COMMAND SYMBOL

FM VALID ONLY FOR OCR INPUT

Format Merge

EXAMPLES of command FM

The Format Merge command was used many times to produce the page you are now reading. Let's look at two instances. Glance at the title FORMAT MERGE at the top of the adjoining column.

Here is the format that does it;
***1st line** issues a "guide mark"used to position the heavy vertical line,*
***2nd line** then sets up the type specs for the head, and waits for the "Format Merge" copy,*
***3rd line** finally changes specs for the "Also" line.*

Format C2:
‹FDc2;it14.6;va40› I ‹ql;it16,0;nr6,16,0›
‹FM› _____ ‹ql;nb2,9,12;FE›

Type copy thus:
‹/c2› FORMAT MERGE **‹FM›** (Also called "Format Interrupt") ‹QL›

The Result is this:

FORMAT MERGE

(Also called "Format Interrupt")

The next example shows the format that handles the paragraph called **COMMAND** *on this page. The merge changes font, size and leading.*

Format PG:
‹FDpg;xx;ve14;cw42;
nr6,12,12;an4;it16› **‹FM›** _____ ‹nr2,
10,11.5;FE›

Type copy thus:
‹/pg› COMMAND ‹FM› is used to exchange control from within a format to ....etc...

The Result is this:

COMMAND is used to exchange co
story file or back again. In essence, the
into segments with "stop" and "go-ba
among commands in that the SAME co

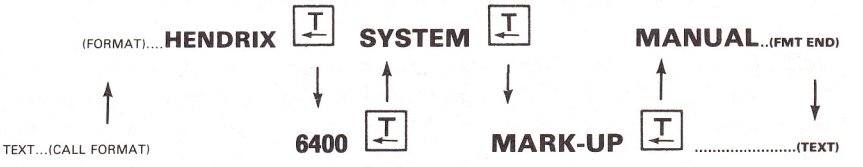
FORMAT MERGE

(Also called "Format Interrupt")

..‹fm›..or.. T .. Single-Key "F. MERGE"

(2-letter command valid only for OCR input)

COMMAND exchanges control from within a format to the story file or back again. In essence, the format (and the text) are broken into segments with "stop" and "go-back" signs in between. It is unique among commands in that the SAME command is used to go in either direction, to the text or to the format.
For instance, take a portion of a format and of some text. Merge points are shown by the symbol: ⏎



EXAMPLE OF ACTION:

1. Format is activated from the textstream by using a name call-out, and it is executed to beyond the word "HENDRIX"where a "format merge point" is found.
2. Program now returns to the original text and reads past "6400" to where the first "text merge point" is issued.
3. Processing goes back to the format, reads "SYSTEM" and then encounters the second "merge".
4. Control is transferred to text until its second "merge" is keyboarded.
5. A return is made to the format. Since there are no more "merges", the remainder of the format is processed. When the end is reached, the format becomes inactive and the program resumes reading the text.

RESULT: **HENDRIX 6400 SYSTEM MARK-UP MANUAL**

EXCEPT If FORMAT MERGE is encountered in the text while format is inactive,

1. it will be considered a QUAD LEFT instead.

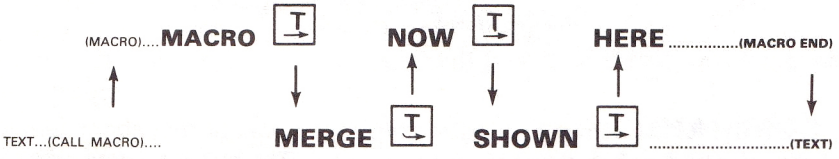
SINGLE KEY COMMAND SYMBOL

MACRO MERGE

(Also called "Macro Call Point")

..〈mm〉..or.. .. **Single-Key "M. MERGE"**

COMMAND is used to transfer control from within a MACRO to the textstream and back again. It functions in the same way as Format Merge, as shown below:



EXAMPLE OF ACTION:

See explanation on opposite page.

RESULT: **MACRO MERGE NOW SHOWN HERE**

EXCEPT: If a MACRO MERGE is read in the textstream and there is no macro active at the time,

1. It will be treated as a FORMAT MERGE and
2. If no format is active it will become a QUAD LEFT.

EXAMPLES of command MM

In order to use a macro merge we first must define the macro itself and insert the merge point somewhere within it.
The 18 point headline format called "H18" contains such a macro designed toproduce a byline.

Format H18:
〈FDh18〉 ..... 〈MD1;fn61,10,16 〉
By **〈MM〉** _____
〈qc;/news;MD〉 ... 〈FE〉

Note above that after the merge there is a quad center followed by an "automatic" call for "news" format (used in page 4-7 sample), before macro is ended.

(• denotes macro call.)
(☐ denotes macro merge.)

Type copy thus:
〈/H18〉 Headline goes here 〈FM〉
•1 Joe Reporter☐ The paragraph you are...

The Result

Headline goes here

By Joe Reporter

The paragraph you are now reading shows how the Hendrix 6400 system accepts a Composition Command, interprets it, calculates, and then issues the necessary instructions to the output typesetter.

SB/SE

String Begin/String End

EXAMPLES of command SB...SE

Suppose we need a running head exactly four times. We set up a string **SB4.....SE** *after the type specs, (All leader dots are for clarity, they are NOT parts of the format) thus:*

⟨fd/HJ ; cw11,7,10,14⟩ ..... ⟨SB4⟩
THE HENDRIX JOURNAL
⟨qc;SE⟩ ..... ⟨fe⟩ .....

Now when we call Format "HJ", it repeats the copy four times.

THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL

By adding another string, the program will jump out of the first and into the second, (printing first 3 and then 5 and so on).

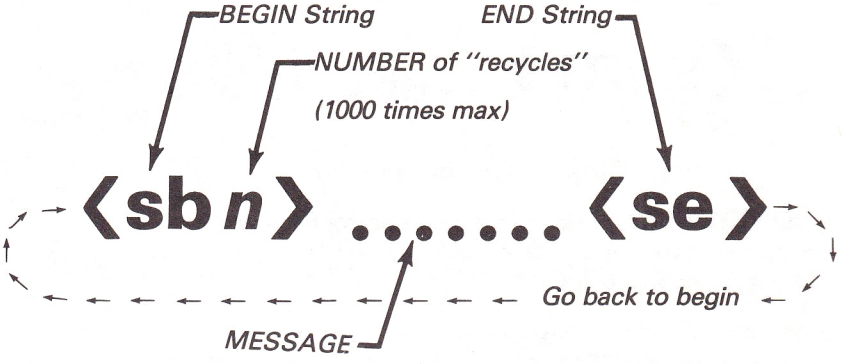
⟨fd/HSP; cw11,7,10,12⟩ ..... ⟨SB3⟩
THE HENDRIX JOURNAL
⟨qc;SE⟩ ..... ⟨ps7,9;SB5⟩ . .....
SPORTS SECTION **⟨qc;SE⟩**
..... ⟨fe⟩

Now call format "HSP"

THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
SPORTS SECTION
SPORTS SECTION
SPORTS SECTION
SPORTS SECTION
SPORTS SECTION

STRING BEGIN/STRING END

(Also called "Message Recycling")



COMMAND allows a string of commands and/or text characters to be positioned and recognized within a format. The string starting point is indicated by STRING BEGIN **⟨sb⟩** command. The STRING END **⟨se⟩** command terminates the string and then sends the program back to starting point of the string.

There are two techniques that are used with strings:

1. Specify the parameter "*n*" in the Begin string command and the string will automatically recycle "*n*" times. The maximum value of "*n*" is 1000.
2. Include one (or more) FORMAT MERGE commands within the string message area. The string will stop at the "format merge point" and wait for text. The "text merge", in turn, continues the string until the end is reached and another cycle is started (or it goes on to the next "merge point, if there are several").

BEGINS whenever the **⟨sb⟩** command is encountered.

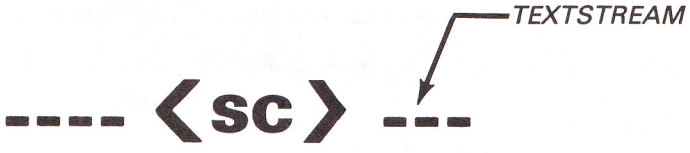
ENDS

1. When the string has recycled "*n*" times. It then automatically goes to the character immediately following the string.
2. If no value is given for "*n*", then the string recycles for 200 times under program control, then exits to the next character after the string end.
3. With a **⟨sc⟩** command. *(See next page).*

SC

STRING CANCEL

(Also called "Stop Recycle")



COMMAND is intended to provide a means of exiting from a repeating string containing a Format Merge. This is the ONLY method of stopping such a string. It also allows the remainder of the format to be processed. The command is issued FROM THE TEXT, in place of a "merge".

BEGINS when the program encounters the STRING CANCEL **‹sc›** in the textstream. A jump is made immediately to the character following the end of the string.

EXCEPT an error message will be output, if no string has been made active and "format merge point" is found.

EXAMPLES of command SC

*Moreover, when we use a Format Merge **within** the string, then **we** control the number of times the string repeats, by the number of times we use a merge command in the textstream, like this:*

‹fd HJ ; cw11,7,12,14› ..... **‹SB›**
HENDRIX JOURNAL-Page
‹FM;qr;SE› ..... **‹fe›**

Type:
‹/HJ› 1 **‹fm›** 3 **‹fm›** 5 **‹fm›** 7 **‹fm›**

HENDRIX JOURNAL—Page 1
HENDRIX JOURNAL—Page 3
HENDRIX JOURNAL—Page 5
HENDRIX JOURNAL—Page 7
HENDRIX JOURNAL—Page

Wait! There's an EXTRA line above that's not needed or wanted. Why? The string looped again and then waited for another page number. The solution is to EXIT with an SC command (instead of a Format Merge) when we're done. To end the last line a Quad Right is also needed. (The format HAD supplied it, AFTER merge.)

Type:
‹/HJ› 21 **‹fm›** 23 **‹fm›** 27 **‹SC;qr›**

HENDRIX JOURNAL—Page 21
HENDRIX JOURNAL—Page 23
HENDRIX JOURNAL—Page 25
HENDRIX JOURNAL—Page 27

GV•

General Variable

EXAMPLES of command GV

Look again at the "string", (on page 4-10) that put out
THE HENDRIX JOURNAL
four-times-in-a-row.

Let's illustrate the use of a General Variable by "controlling" the number of times this particular string will recycle.

(1) First, we rewrite the string using a "T" to stand for the number of TIMES the string is to be used. When enclosed in parenthesis, (T) is the call-out for our General Variable. The string itself now appears this way:

..... **‹SB (T)›**
THE HENDRIX JOURNAL
‹qc;SE› .....

(2) Before we can use this string , we must first define what "T" shall be. Then we can call the format containg the string, thus:

‹gvT:5›
‹/HJ›
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL

Note above that five lines were automatically output.

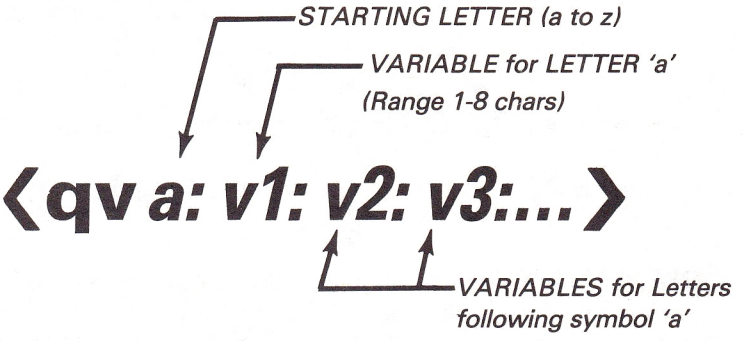
(3) If we change the General Variable value from 5 to 3, the results change accordingly:

‹gvT:3›
‹/HJ›
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL
THE HENDRIX JOURNAL

There's more on the next page.

GENERAL VARIABLE

(Also Called Equivalent Values)



COMMAND defines string(s) of up to eight characters to be equated to a single alpha symbol (a-z). The symbol itself is then used to replace a command or parameters within a format. When the format becomes active the actual definitions will be substituted for the symbols. Let's use an example to explain this command.

Assume three different point sizes ("X, Y and Z") are used in a series of formats used for grocery ads, then general variables starting with X:

‹gv *X:18:12:24* ›

yields "X" equal to 18 pts..."Y" equal to 12 pts..."Z" equal to 24 pts. By redefining the point size values to be:

‹gv *X:28:14:36* ›

all of the grocery formats now have a new combinations of point sizes.

STORES as soon as a definition is read in. A general variable can be defined and stored anytime and is especially useful when contained within a format. By calling different formats, a great variety of general variables can be redefined with a new set of values. Formats thus can do double-duty, triple-duty or more!

EXCEPT

1. No more than eight (8) characters between colons (:) are allowed or an error message will show up.
2. No more than 26 general variables may exist. Either a lower case (a-z) or an upper case (A-Z) may be used for any particular symbol.

GENERAL VARIABLE CALL

(Also called "Get Variable")

(a) or (A)

CALL-OUT for General Variable provides a symbol to be used in formats in place of a previously defined string. It is composed of an alphabetic character ENCLOSED IN PARENTHESIS.

USED in formats ONLY. If employed in the textstream, by mistake, it will be read as printable characters, not as a call-out.

BEGINS when format is active.

EXCEPT if the symbol has not been defined by a **‹gv›** command, the call-out will be ignored. It will NOT cause an error message to appear.

EXAMPLES of command GV and CALL

(continued from prior page)

(1) A practical application is to control the number of Date Lines and Folios. Say we have a 12 page newspaper, and "f" is the number of folios.

‹gvf:12›
‹/DATELN›

HENDRIX MONITOR, Friday, March 11 — 2
3 — HENDRIX MONITOR, Friday, March 11
HENDRIX MONITOR, Friday, March 11 — 4
5 — HENDRIX MONITOR, Friday, March 11
HENDRIX MONITOR, Friday, March 11 — 6
7 — HENDRIX MONITOR, Friday, March 11
HENDRIX MONITOR, Friday, March 11 — 8
9 — HENDRIX MONITOR, Friday, March 11
HENDRIX MONITOR, Friday, March 11 — 10
11 — HENDRIX MONITOR, Friday, March 11
HENDRIX MONITOR, Friday, March 11 — 12

HENDRIX

TECHNICAL DOCUMENTATION
USERS' COMMENT FORM

TITLE:

DOCUMENT NO:

REVISION NO:

ERRORS NOTED:

SUGGESTIONS FOR IMPROVEMENT:

FROM: NAME _____ DATE _____

COMPANY _____

TITLE _____

ADDRESS _____

_____

FIRST CLASS
PERMIT NO. 6782
MANCHESTER
N.H.

BUSINESS REPLY MAIL

No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

HENDRIX
645 HARVEY ROAD
MANCHESTER, NH 03103

ATTN: Technical Publications Dept.

HENDRIX

HENDRIX 6400 COMPOSITION SYSTEM
SUMMARY OF COMMANDS

2-LETTER CODE	COMMAND DESCRIPTION	COMMAND & PARAMETERS	CANCEL CMND
AA	ALLOW AUTOMATIC para. indent	⟨aa⟩	XA
AD	ALLOW DICTIONARY mode	⟨ad⟩	XD
AE	ALLOW ESTIMATOR for heads	⟨ae⟩	XE
AF	ALLOW FLASH	⟨af⟩	XF
AH	ALLOW HYPHENATION mode	⟨ah⟩	XH
AL	ALLOW LETTERSPACING mode	⟨al⟩	XL
AM	ALTER MAXIMUM spaceband	⟨am⟩	XM
AN	ALTER MINIMUM spaceband	⟨an⟩	XN
AW•	ALTER WIDTH	⟨aw*u or -u*⟩	XW
CW•	COLUMN WIDTH	⟨cw*p.p*⟩	—
CW•	"FOUR-IN-ONE" command	⟨cw*p,p,p,p,t,p,f*⟩ ...	—
DH•	DEFINE HYPHENATION mode	⟨dh*c,l*⟩	—
[1]EM	EM fixed space	⟨em⟩	—
[1]EN	EN fixed space	⟨en⟩	—
[1]FB	FONT BOLD	⟨fb⟩	—
FD•	FORMAT DEFINE	⟨fd*f.name*⟩	FX
FE	FORMAT END	⟨fe⟩	—
[1]FL	FONT LIGHT	⟨fl⟩	—
[2]FM	FORMAT MERGE	⟨fm⟩	—
FN•	FONT NUMBER	⟨fn*n*⟩	—
GD•	GALLEY DEFINE	⟨gd*w,d,g,n,e*⟩ ..	GX
GE	GALLEY END point	⟨ge⟩	—
GF	GALLEY FREE (discretionary end)	⟨gf⟩	—
GV•	GENERAL VARIABLE define	⟨gv*a:v1,v2,..*⟩ ..	—
HH•	HANG HERE	⟨hh*n*⟩	HX
HI•	HANG INDENT	⟨hi*l.l,r.r*⟩	HX
HO	HOMOGRAPH (discret. hyphen)	⟨ho⟩	—
HP•	HORIZONTAL space in PICAS	⟨hp*p.p*⟩	—
HR	HANG REPEAT indent	⟨hr⟩	HX
HU•	HORIZONTAL space in UNITS	⟨hu*u*⟩	—
IH	INDENT HANG copy	⟨ih⟩	IR
IP•	INDENT PARAGRAPH	⟨ip*l.l,r.r*⟩	QP
IT•	INTENT TAKE	⟨it*l.l,r.r*⟩	IX
[1]JU	JUSTIFY LINE	⟨ju⟩	—
LL•	LENGTH fit LEFT	⟨ll*l.l,d.d*⟩	LX
LR•	LENGTH fit RIGHT	⟨lr*r.r,d.d*⟩	LX
[2]MC	MACRO CALL	⟨mc⟩	—
[2]MD	MACRO DEFINE	⟨md*n*⟩	—
[2]MM	MACRO MERGE	⟨mm⟩	—
NC	NO COUNT line	⟨nc⟩	—
PC•	PHOTO COMP drive & font select	⟨pc*n,a,b,c,..*⟩ ..	—
PI•	PI character (and font) call	⟨pi⟩ *a* ⟩	—
PS•	POINT SIZE	⟨ps*p.t*⟩	—
PW•	POINT WIDTH	⟨pw*p.t*⟩	—
[1]QC	QUAD CENTER	⟨qc⟩	—
[1]QL	QUAD LEFT	⟨ql⟩	—
[1]QM	QUAD MIDDLE	⟨qm⟩	—
[1]QP	QUAD PARAGRAPH	⟨qp⟩	—
[1]QR	QUAD RIGHT	⟨qr⟩	—
RB	RAIL BOLD	⟨rb⟩	—
RC	RAGGED CENTER	⟨rc⟩	RX
RI	RAIL ITALIC	⟨ri⟩	—
RL	RAGGED LEFT	⟨rl⟩	RX
RR	RAGGED RIGHT	⟨rr⟩	RX
SB•	STRING BEGIN (in format)	⟨sb*n*⟩	SC
SE	STRING END (in format)	⟨se⟩	SC
SL•	SKEW LEFT indent	⟨sl*l.l,d.d*⟩	SC
SR•	SKEW RIGHT indent	⟨sr*r.r,d.d*⟩	SC
ST	STRIP TAKE compose	⟨st⟩	—
TA	TAB ACTIVATE mode	⟨ta⟩	TX
TS•	TAB STOP	⟨ts*i,s1,s2,...*⟩ ...	TX
TI•	TAB INDENTeach column	⟨ti*l.l,d.d*⟩	TC
[1]TH	THIN fixed space	⟨th⟩	—
TK	TAB KILL remaining columns	⟨tk⟩	—
TE•	TAB EQUAL number of columns	⟨te*i,n*⟩	TX
TP•	TAB PROPORTIONAL each col.	⟨tp*i,p1,p2,...*⟩ ..	TX
TT	TAB TEXT	⟨tt⟩	TX
TW•	TAB WIDTH	⟨tw*i,w1,w2,..*⟩ ...	TX
TV	TAB VARIABLE space	⟨tv⟩	TX
VA•	VERTICAL ADVANCE leading	⟨va*p.f*⟩	—
VB•	VERTICAL BACK leading	⟨vb*p.f*⟩	—
VC	VERTICAL CLEAR leading counter	⟨vc⟩	—
VE•	VERTICAL EXTRA leading	⟨ve*p.f*⟩	—
VI	VISUALS START on tape	⟨vi⟩	XV
VL•	VERTICAL LEADING	⟨vl*p.f*⟩	—
VM	VERTICAL MARK	⟨vm⟩	—
VR	VERTICAL RETURN to mark	⟨vr⟩	—
VU•	VERTICAL UNIT leading	⟨vu*u*⟩	—
WB	WUN-LINE BOLD	⟨wb⟩	—
WF•	WUN-LINE FONT change	⟨wf*n*⟩	—
WL	Fill WITH LEADERS	⟨wl⟩	—
WR	Fill WITH RULES	⟨wr⟩	—
WX	Fill WITH character 'X'	⟨wx⟩	—
WZ	WUN-LINE ZERO leading	⟨wz⟩	—
XX	CANCEL ALL Tabs/Indents	⟨xx⟩	—

Special PACESETTER Commands

IE	INITIALIZE EIGHT	⟨ie⟩	—
IN	INITIALIZE NINE	⟨in⟩	—

Special CG UNIVERSAL Commands

NR•	NORMAL ROMAN	⟨nr*g*⟩	—
NI•	NORMAL ITALIC	⟨ni*g*⟩	—
NB•	NORMAL BOLD	⟨nb*g*⟩	—
ND•	NORMAL BOLD ITALIC	⟨nd*g*⟩	—
CR•	CONDENSED ROMAN	⟨cr*g*⟩	—
CI•	CONDENSED ITALIC	⟨ci*g*⟩	—
CB•	CONDENSED BOLD	⟨cb*g*⟩	—
CD•	CONDENSED BOLD ITALIC	⟨cd*g*⟩	—
ER•	EXPANDED ROMAN	⟨er*g*⟩	—
EI•	EXPANDED ITALIC	⟨ei*g*⟩	—
EB•	EXPANDED BOLD	⟨eb*g*⟩	—
ED•	EXPANDED BOLD ITALIC	⟨ed*g*⟩	—
NV	NEGATIVE VIDEO	⟨nv⟩	PV
PV	POSITIVE VIDEO	⟨pv⟩	NV

Special Autologic APS-4/5 Command

SO•	SET OBLIQUE	⟨so*a*⟩	XO

[1] Has a more convenient single key form.
[2] Use single-key command. The 2-letter command valid only for OCR.

HENDRIX
WORLDWIDE